

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a biomedicínského inženýrství**

**Výukový kybernetický model s krokovými  
motory - laboratorní úloha**

**Educational cybernetic model with stepper  
motors - laboratory exercise**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

## Zadání bakalářské práce

Student: **Adam Trefil**  
Studijní program: B2649 Elektrotechnika  
Studijní obor: 3901R039 Biomedicínský technik  
Téma: **Výukový kybernetický model s krokovými motory - laboratorní úloha**  
**Educational Cybernetic Model with Stepper Motors**  
**- Laboratory Exercise**

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Seznámení se s konstrukcí krokových motorů a způsoby jejich řízení, zejména v biomedicínských aplikacích.
2. Studium návrhových metod FPGA logiky a způsobu komunikace s modulem PmodSTEP.
3. Návrh koncepce výukového kybernetického modelu s krokovými motory a měřením polohy.
4. Návrh obvodové logiky laboratorní úlohy s modulem PmodSTEP pro vývojovou desku s FPGA.
5. Realizace fyzikálního výukového modelu s krokovými motory.
6. Implementace, oživení a ověření funkce navržené logiky pro FPGA.
7. Vytvoření laboratorní úlohy pro řízení vytvořeného kybernetického modelu.
8. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] ŘEZÁČ, Kamil. Krokové motory : princip funkce, metody řízení. In: *Robotika.cz* [online]. 2002-10-28. [cit. 2017-06-30]. Dostupné z: <https://robotika.cz/articles/steppers/cs>.
- [2] PARNELL, Karen and Nick MEHTA. *Programmable Logic Design Quick Start Handbook*. 4th ed. [s.l.]: Xilinx Inc., 2003. 225 s.
- [3] ASHENDEN, Peter J. *The Designer's Guide to VHDL*. San Francisco (USA): Morgan Kaufmann Publishers, 1999. 688 s. ISBN 1-55860-270-4.
- [4] KAŠÍK, Vladimír. *Programování hradlových polí*. Učební text a návody do cvičení. Ostrava: VŠB-TU Ostrava, 2012.
- [5] ŠTASTNÝ, Jakub. *FPGA prakticky*. 1. vyd. Praha: BEN - technická literatura, 2010. 199s. ISBN 978-80-7300-261-9.
- [6] DIGILENT, Inc. *Pmod STEP: Stepper Motor Driver* [online] Dostupné z: <http://store.digilentinc.com/pmod-step-stepper-motor-driver/>.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

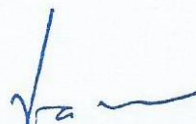
Vedoucí bakalářské práce: **Ing. Vladimír Kašík, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jiří Koziorek, Ph.D.  
*vedoucí katedry*




prof. Ing. Pavel Brandštetter, CSc.  
*děkan fakulty*



## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum: 30. dubna 2018

  
.....

podpis studenta

## **Poděkování**

Rád bych poděkoval vedoucímu mé bakalářské práce, panu Ing. Vladimíru Kašíkovi, Ph.D., za odborné vedení, cenné rady, připomínky a konzultace, které mi pomohly zpracovat tuto bakalářskou práci.

## **Abstrakt**

Cílem této bakalářské práce je snímání hodnot akcelerometru na modulu PmodACL2. Z výsledných hodnot je vypočítán klouzavý průměr a následně v IP jádře Cordic vypočítán úhel natočení. Tento úhel odpovídá úhlu otočení krokového motoru ovládaný modulem PmodSTEP. Ke hřídeli krokového motoru je upevněn model ruky, která zrcadlí pohyb paže uživatele. Práce slouží jako možný podklad pro laboratorní úlohy, v příloze se nachází dvě laboratorní úlohy, jedna jednodušší a druhá složitější.

## **Klíčová slova**

FPGA; Nexys 3; krokový motor; akcelerometr; PmodACL2; PmodSTEP; telemedicína; Cordic.

## **Abstract**

The aim of the bachelor thesis is sensing values from accelerometer in the module PmodACL2. Moving average is calculated from the result values and angle of rotation is calculated by the IP core. This angle corresponds to the rotation of the stepper motor with the PmodSTEP module. The shaft of stepper motor is placed on the model of hand, and the hand model copies the movement of user's hand. This thesis can be used as a laboratory exercise. The thesis contains two examples of possible laboratory exercises with different difficulty level.

## **Key words**

FPGA; Nexys 3; stepper motor; accelerometer; PmodACL2; PmodSTEP; telemedicine; Cordic.

# Obsah

Seznam použitých symbolů a zkratk	9
Seznam obrázků	11
Seznam tabulek	12
Úvod	13
<b>1 Konstrukce krokových motorů</b>	<b>14</b>
1.1 Princip krokového motoru . . . . .	15
1.2 Způsob řízení krokového motoru . . . . .	15
1.2.1 Unipolární a bipolární řízení . . . . .	15
1.2.2 Jednofázové versus dvoufázové řízení . . . . .	16
1.2.3 Řízení s plným a polovičním krokem . . . . .	16
1.2.4 Řízení krokového motoru . . . . .	16
<b>2 Programovatelné logické obvody</b>	<b>19</b>
2.1 Jednoduchý programovatelný logický obvod . . . . .	19
2.2 Komplexní programovatelný logický obvod . . . . .	19
2.3 Programovatelná hradlová pole . . . . .	20
2.3.1 Rozdíl mezi ASIC a FPGA . . . . .	21
2.3.2 Aplikace . . . . .	22
<b>3 Způsoby návrhu logiky FPGA</b>	<b>24</b>
3.1 VHDL . . . . .	24
<b>4 Vývojová deska Nexys 3</b>	<b>25</b>
<b>5 Rozhraní Pmod</b>	<b>26</b>
5.1 Modul PmodACL2 . . . . .	27
5.2 Modul PmodSTEP pro řízení krokových motorů . . . . .	28
<b>6 Návrh koncepce výukového kybernetického modelu s krokovými motory a měřením polohy</b>	<b>30</b>
6.1 Výběr vhodného krokového motoru . . . . .	30
<b>7 Návrh obvodové logiky laboratorní úlohy s modulem PmodSTEP pro vývojovou desku s FPGA</b>	<b>32</b>
7.1 Komponenty návrhu . . . . .	32
7.2 Komponenta Signal_START . . . . .	33
7.3 Komunikace s akcelerometrem . . . . .	34
7.4 Komponenta ACL_SPI . . . . .	36
7.4.1 SPI_control . . . . .	36

7.4.2	data_transmission . . . . .	37
7.4.3	slave_connection . . . . .	37
7.5	Komponenta data_processing_component . . . . .	37
7.6	Komponenta arctg . . . . .	38
7.7	Komponenta data_evaluation . . . . .	40
<b>8</b>	<b>Realizace fyzikálního výukového modelu s krokovými motory</b>	<b>43</b>
<b>9</b>	<b>Implementace, oživení a ověření funkce navržené logiky pro FPGA</b>	<b>44</b>
9.1	Ověření funkčnosti a zhodnocení výsledků . . . . .	45
	<b>Závěr</b>	<b>48</b>
	<b>Seznam použité literatury</b>	<b>49</b>
	<b>Seznam příloh</b>	<b>52</b>



## Seznam použitých symbolů a zkratek

A/D	Analogově digitální převodník
ADC	Analog Digital Converter
ASIC	Application Specific Integrated Circuits
CLB	Konfigurovatelné logické bloky
CLK	Hodinový signál clock
CMOS	Complementary Metal–Oxide–Semiconductor
CPLD	Complex Programmable Logic Device
DPS	Deska plošných spojů
DSP	Digitální signálový procesor
EEPROM	Electrically Erasable PLD
EPLD	Electrical PLD
FIFO	First In First Out
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
GAL	Generic Array Logic
GPIO	General-purpose input/output
HDL	Hardware Description Language
HŘD	Hostitelská řídicí deska
HW	Hardware
I2C	Inter-Integrated Circuit
I/O	Input/Output
IP core/jádro	Intellectual Property Core/jádro
LCA	Logic Cell Array
LUT	Look-up tabulka
LSB	Least Significant Bit
MEMS	Micro–Electro Mechanical System
MISO	Master In Slave Out
MOSI	Master Out Slave In
OTP FPGA	One-time programmable FPGA
PAL	Programmable Array Logic
PLA	Programmable Logic Arrays
PLD	Programovatelný logický obvod
PM	Permanentní magnet
Pmod	Peripheral Module interface
RAM	Random access memory
RF	Rádiová frekvence
RST	Reset
RTL	Register-transfer level

SAN	Storage Area Network
SCL	Synchronous Clock
SCLK	Serial clock
SDA	Synchronous Data
SDI	Serial Data Input
SDO	Serial Data Output
SoC	System on Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SS	Slave Select
SW	Software
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VR	Variabilní reluktance
XPGA	eXpanded Programmable Gate Array

## Seznam obrázků

1	Celkový pohled na krokový motor . . . . .	14
2	Stator . . . . .	14
3	Rotor . . . . .	14
4	Schéma unipolárního řízení krokového motoru . . . . .	15
5	Schéma bipolárního řízení krokového motoru . . . . .	16
6	Rozložení cívek krokového motoru . . . . .	16
7	Typy PLD . . . . .	19
8	CPLD architektura . . . . .	20
9	FPGA architektura . . . . .	21
10	Vývojová deska Nexys 3 . . . . .	25
11	Pmod hostitelský konektor . . . . .	26
12	Rozhraní Pmod . . . . .	26
13	Funkční diagram akcelerometru ADXL362 . . . . .	28
14	Modul PmodSTEP . . . . .	29
15	Blokové schéma návrhu výukového modelu . . . . .	30
16	Diagram připojení periferie k FPGA . . . . .	32
17	Blokové schéma děličky hodinového signálu . . . . .	33
18	Část kódu znázorňující děličku hodinového signálu . . . . .	34
19	Komunikace přes SPI . . . . .	34
20	Časový diagram zápisu do registru modulu PmodACL2 . . . . .	35
21	Časový diagram čtení z registru modulu PmodACL2 . . . . .	36
22	Fragment VHDL kódu pro omezení kmitů paže . . . . .	38
23	Diagram principu klouzavého průměru . . . . .	38
24	Pohled z boku na modul PmodACL2 . . . . .	39
25	Znázorněn výpočet úhlu . . . . .	39
26	Fragment VHDL kódu pro tlačítko . . . . .	41
27	Část VHDL kódu pro děličku hodinového signálu . . . . .	41
28	Diagram procesu krokování . . . . .	42
29	VHDL kód pro vyhodnocení kroků . . . . .	42
30	Celkový pohled na model . . . . .	43
31	Detail zapojení PmodSTEP s krokovým motorem . . . . .	43
32	Implementace . . . . .	44
33	Symbols, a) „fajfka“, b) varování, c) chyba . . . . .	44
34	Program Digilent Adept . . . . .	45
35	Detailní zobrazení prvního zápisu do registru akcelerometru . . . . .	46
36	Průběh čtení dat z akcelerometru . . . . .	46
37	Příkaz pro čtení z akcelerometru . . . . .	47
38	Detailní zobrazení příkazu pro čtení z akcelerometru . . . . .	47

## Seznam tabulek

1	Unipolární jednofázové řízení s plným krokem . . . . .	17
2	Unipolární dvoufázové řízení s plným krokem . . . . .	17
3	Unipolární řízení s polovičním krokem . . . . .	17
4	Bipolární jednofázové řízení s plným krokem . . . . .	17
5	Bipolární dvoufázové řízení s plným krokem . . . . .	18
6	Bipolární řízení s polovičním krokem . . . . .	18
7	Popisy pinů konektoru J1 . . . . .	27
8	Popisy pinů konektoru J2 . . . . .	28
9	Popisy pinů konektoru J3 . . . . .	28
10	Rozdělení pinů modulu PmodSTEP . . . . .	29
11	Porovnání různých dostupných krokových motorů . . . . .	31
12	Přehled I/O pinů entity kyberneticky_model . . . . .	33
13	Konfigurace ACL . . . . .	35
14	Datové registry modulu PmodACL2 . . . . .	36
15	Vstupy a výstupy komponenty arctg . . . . .	39
16	1Q7 formát dat . . . . .	39
17	2Q6 formát dat . . . . .	39
18	Přehled I/O signálů komponenty data_evaluation . . . . .	40

## Úvod

Biomedicína se v posledních letech vyvíjí ohromnou rychlostí, dochází ke zlepšování a modernizaci přístrojů, přičemž je kladen velký důraz na vývoj softwaru, který je během studia více upřednostňován oproti hardwaru. Z tohoto důvodu je téma mé práce zaměřeno na krokové motory.

Cílem bakalářské práce je popsání technologie, návrhu a samotného vytvoření kybernetického modelu poháněného krokovým motorem.

Teoretická část práce obsahuje obecné seznámení s konstrukcí krokových motorů a způsoby jejich řízení. Práce stručně nastíní technologii PLD, CPLD a FPGA. V dalších kapitolách se popisuje princip použití a případné dělení technologie. Dalším bodem je způsob návrhu logiky FPGA a možné varianty řešení, a to s pomocí programovacích jazyků Verilog a VHDL, popisujících hardware zařízení.

Obsahem praktického řešení mé práce je popis parametrů jednotlivých komponent. Nejprve popis parametrů, periférií a využití vývojové desky Nexys 3 od firmy Digilent, dále seznámení s rozhraním Pmod, jeho možnostmi a variantami provedení. Dále je obsažen popis konkrétních modulů, které využívají zmíněného rozhraní jako PmodACL2 a PmodSTEP a principů komunikace s vývojovou deskou skrze toto rozhraní. Následuje návrh koncepce výukového modelu s krokovým motorem a měřením polohy, kde bude seznámení se s představou celkového modelu, zapojení zmíněných modulů, krokového motoru a robotického ramene. Dalším bodem je návrh obvodové logiky, ve kterém bude popsána hierarchie návrhu a funkce jednotlivých komponent. Návrh koncepce výukového modelu s krokovým motorem a měřením polohy tvoří jednu z hlavních částí práce. Nedílnou součástí je implementace, oživení logiky a následné ověření funkčnosti modelu. V závěru se nachází zhodnocení dosažených výsledků práce. Poslední částí bude zpracování laboratorní úlohy, ve které bude obsažen cíl úlohy, zadání, teoretický rozbor a pracovní postup.

## 1 Konstrukce krokových motorů

Krokový motor se skládá ze statického statoru a pohyblivého rotoru. Rotor se dále skládá z hřídele usazené na kuličkových ložiscích a prstence permanentních magnetů. Stator se skládá ze sady cívek, kde pólové nástavce jsou vroubkovány se stejnou roztečí, jako je rozteč magnetů na rotoru. Díky tomu získáme vyšší přesnost motoru při stejném počtu cívek.[1]



Obrázek 1: Celkový pohled na krokový motor[1]



Obrázek 2: Stator[1]



Obrázek 3: Rotor[1]

## 1.1 Princip krokového motoru

Základním principem krokového motoru je procházení proudem cívkou statoru, díky čemu se vytvoří magnetické pole, které přitáhne opačný pól magnetu rotoru. Zapojováním různých cívek dosáhneme rotačního pohybu hřídele rotoru.

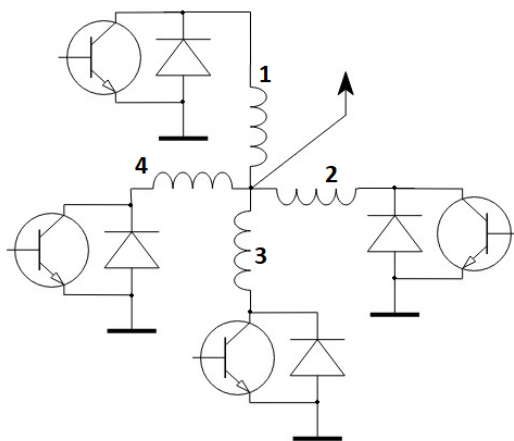
Přechodové magnetické jevy ovlivňují maximální rychlost otáčení motoru (stovky kroků za sekundu, záleží podle konkrétního motoru). Při překročení této rychlosti nebo při velké zátěži dojde ke ztrátě kroků. Existuje několik variant řízení krokového motoru, avšak nejprve je potřeba si určit požadovaný kroutící moment, přesnost nastavení polohy (kolik stupňů je jeden krok) a odběr proudů. Podle materiálu rotoru se krokové motory rozdělují na několik typů, buď může být tvořen železným jádrem tzv VR (variabilní reluktance), permanentními magnety (PM) a nebo může být hybridní, což je kombinace VR a PM.[1][2]

## 1.2 Způsob řízení krokového motoru

Existuje několik variant řízení motoru, různé způsoby jsou vypsány v následujících kapitolách.

### 1.2.1 Unipolární a bipolární řízení

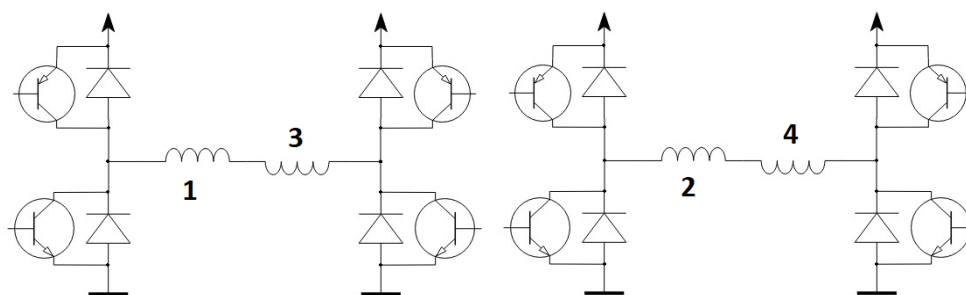
Unipolární řízení je specifické tím, že proud prochází v jednom okamžiku pouze jednou cívkou a motor s tímto buzením má nejmenší odběr, ale také má nejmenší kroutící moment. Výhodou tohoto řízení je jednoduché zapojení (stačí jen jeden tranzistor na každou cívku).



Obrázek 4: Schéma unipolárního řízení krokového motoru[1]

Naproti tomu bipolární řízení se vyznačuje tím, že proud prochází vždy dvěma protilehlými cívkami. Cívky jsou zapojeny tak, že mají opačnou polaritu. Tímto zapojením dosáhneme většího kroutícího momentu. Nevýhodou je vyšší spotřeba (krokový motor se i více zahřívá). Náročnější je také i zapojení, jsou zapotřebí 2 H-můstky (pro každou větev jeden).[1]





Obrázek 5: Schéma bipolárního řízení krokového motoru[1]

### 1.2.2 Jednofázové versus dvoufázové řízení

Jednofázové řízení spočívá v tom, že magnetické pole generuje pouze jedna cívka (při unipolárním buzení) případně dvojice cívek (při bipolárním buzení).

Dvoufázové řízení funguje tak, že dvě sousední cívky mají stejně orientované magnetické pole. Získáme tím vyšší krouticí moment a dvojnásobnou spotřebu oproti jednofázovému řízení.[1]

### 1.2.3 Řízení s plným a polovičním krokem

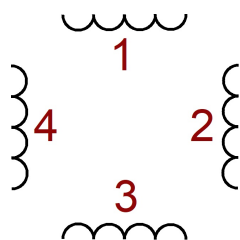
Řízení s plným krokem je takový typ řízení, kdy se provede na jednu otáčku tolik kroků, kolik zubů má stator daného motoru. Tohoto krokování dosáhneme použitím jak unipolární tak bipolární metodou buzení.

U řízení s polovičním krokem se jedná o střídání kroků s jednofázovým a dvoufázovým řízením, čímž dosáhneme dvojnásobné přesnosti.[1]

### 1.2.4 Řízení krokového motoru

Pro zjednodušení budeme uvažovat o krokovém motoru se čtyřmi kroky na otáčku.

Černou barvou je označena cívka, která je bez proudu (v tabulkách je zaznačena jako „0“). Modrá cívka přitahuje červený konec magnetu rotoru (v tabulkách je označena jako „-“). Magnetické pole červené cívky přitahuje modrý konec magnetu (v tabulkách označené jako „+“).[1]



Obrázek 6: Rozložení cívek krokového motoru[1]

Tabulka 1: Unipolární jednofázové řízení s plným krokem[1]

Cívka 1	-	0	0	0
Cívka 2	0	-	0	0
Cívka 3	0	0	-	0
Cívka 4	0	0	0	-

Tabulka 2: Unipolární dvoufázové řízení s plným krokem[1]

Cívka 1	-	0	0	-
Cívka 2	-	-	0	0
Cívka 3	0	-	-	0
Cívka 4	0	0	-	-

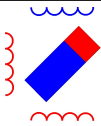
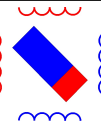
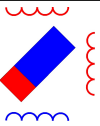
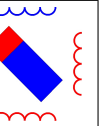
Tabulka 3: Unipolární řízení s polovičním krokem[1]

Cívka 1	-	-	0	0	0	0	0	-
Cívka 2	0	-	-	-	0	0	0	0
Cívka 3	0	0	0	-	-	-	0	0
Cívka 4	0	0	0	0	0	-	-	-

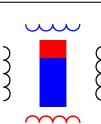
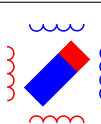
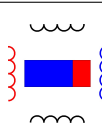
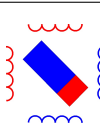
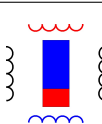
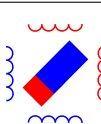
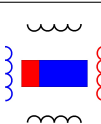
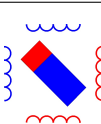
Tabulka 4: Bipolární jednofázové řízení s plným krokem[1]

Cívka 1	-	0	+	0
Cívka 2	0	-	0	+
Cívka 3	+	0	-	0
Cívka 4	0	+	0	-

Tabulka 5: Bipolární dvoufázové řízení s plným krokem[1]

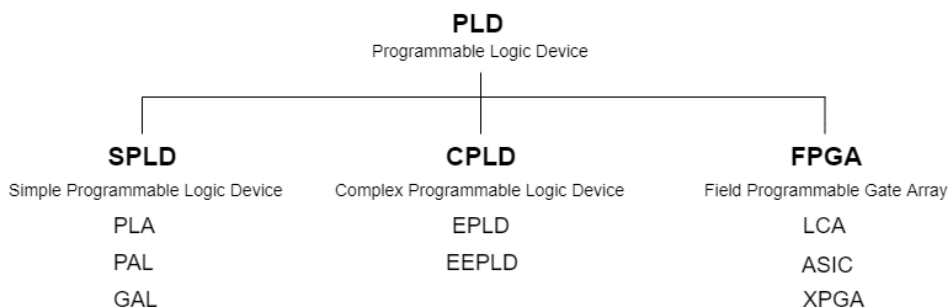
				
Cívka 1	-	+	+	-
Cívka 2	-	-	+	+
Cívka 3	+	-	-	+
Cívka 4	+	+	-	-

Tabulka 6: Bipolární řízení s polovičním krokem[1]

								
Cívka 1	-	-	0	+	+	+	0	-
Cívka 2	0	-	-	-	0	+	+	+
Cívka 3	+	+	0	-	-	-	0	+
Cívka 4	0	+	+	+	0	-	-	-

## 2 Programovatelné logické obvody

Programovatelné logické obvody (PLD) jsou obvody, jejichž funkci můžeme libovolně měnit (programovat). Naprogramováním mikrokontroléru dojde ke změně softwaru, na rozdíl od PLD, kde se změní vnitřní struktura hardwaru. Zjednodušeně si to můžeme představit jako velké množství logických hradel, ze kterých lze naprogramováním složit libovolný digitální obvod (jak kombinační, tak sekvenční).[3][4]



Obrázek 7: Typy PLD[5]

### 2.1 Jednoduchý programovatelný logický obvod

Jedná se o jednoduché programovatelné logické zařízení (SPLD z anglického simple programmable logic device). Výhodou tohoto zařízení je jednoduchost, velikost a cena. Vytvoření logiky se uskutečňuje pomocí rovin (polí) a propojení mezi nimi, roviny jsou AND a OR. A dle možností se SPLD rozděluje na:

**PLA** - Programmable Logic Arrays, programovatelná logická pole. Jsou programovatelné obě roviny AND a OR.

**PAL** – Programmable Array Logic, programovatelná pole logiky. Zde je rovina AND programovatelná, ale rovina OR je pevná.

**GAL** - Generic Array Logic, Generická logická pole. Je to podobná technologie jako PAL, ale může být přeprogramována.[6][7]

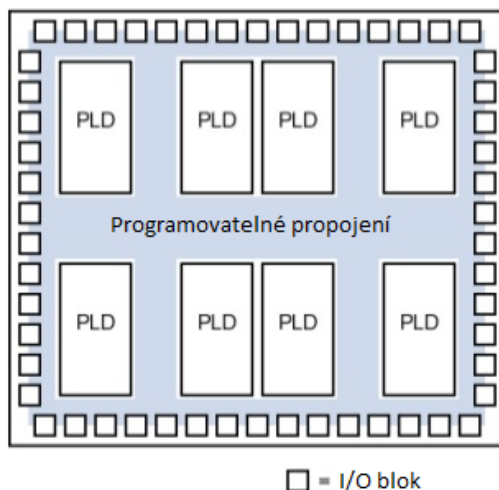
### 2.2 Komplexní programovatelný logický obvod

Complex Programmable Logic Device (CPLD), jsou obvody s menším stupněm integrace a jsou vhodné pro méně složité aplikace. Používají se převážně pro kombinační logiku. Sekvenční logiku můžeme v CPLD realizovat, ale výsledný model zabírá velké množství stavebních bloků. Existuje několik variant, například EEPLD a EPLD, ty se často využívají pro testování návrhu logiky.

**EPLD** - Electrical PLD. Jedná se o integrovaný logický obvod, který obsahuje několik obvodů PLD a je na uživateli, jak je mezi sebou propojí.

**EEPLD** - Electrically Erasable PLD. Je to PLD, které používá paměť EEPROM k uložení logických návrhů.

K popisu hardwarové struktury PLD a CPLD obvodů slouží v dnešní době vyšší programovací jazyky. Těch existuje několik, jmenujme například Verilog, ABEL nebo VHDL.[7][8]



Obrázek 8: CPLD architektura[9]

### 2.3 Programovatelná hradlová pole

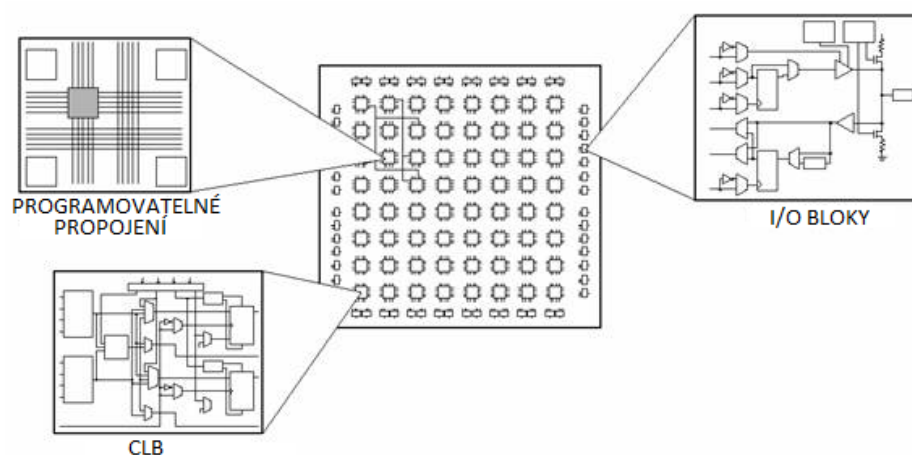
Field-Programmable Gate Array (FPGA) patří mezi PLD obvody, které jsou svou vnitřní strukturou mnohem složitější a velmi se liší od CPLD obvodů. Jsou vhodné pro realizaci sekvenčních logických obvodů, které mohou být složité i jako celé mikroprocesory nebo i digitální filtry. Existuje i jednorázové FPGA (OTP), ale více se používají typy založené na SRAM, které lze během vývoje návrhu přeprogramovat.

Základním stavebním prvkem FPGA jsou konfigurovatelné logické bloky (CLB). Ty obsahují několik slice a ty obsahují logické buňky (logic cell), z nich každá obsahuje look-up tabulku (LUT), klopný obvod typu D a někdy také úplnou sčítačku, případně další bloky.[4]

Moderní FPGA obvody obsahují mimo CLB také několik bloků paměti RAM, bloky pro digitální signálový procesor (DSP), sčítačky, násobičky, generátor hodinového signálu o různých frekvencích atd. Nejnovější FPGA mohou pracovat při frekvenci 500 MHz a více. FPGA mají oproti CPLD obvodům volantní paměť, takže po odpojení napájení dojde k vymazání naprogramované struktury. To znamená, že je nutné po každém zapnutí obvod nejprve naprogramovat, což se většinou realizuje z externí paměti FLASH. Během vývoje lze obvody programovat také přes JTAG rozhraní.

Pro lepší představu si vezmeme FPGA rodinu Spartan 3A typ XC3S200A od firmy Xilinx. Tento obvod patří mezi jednodušší a levnější obvody této firmy. Obsahuje 200 000 hradel, což je 1584 logických buněk. K dispozici tedy je 1584 LUT a D klopných obvodů, které jsou strukturované do „sliců“ a do CLB. V takovémto obvodu lze realizovat jednoduchý procesor s velmi omezenou instrukční sadou a paměť pro program může být přímo v blokové RAM, kterou má tento obvod o velikosti 288 kb.

Když se podíváme na high-endové FPGA (jako je např. Virtex od firmy Xilinx), tak mohou obsahovat milióny buněk a Mb paměti RAM. V takových obvodech je možné realizovat i složité procesory s náročnými aplikacemi jako jsou například řízení A/D převodníku nebo CIC filtr spojený s procesorem.[10]



Obrázek 9: FPGA architektura[11]

#### Mezi FPGA se řadí:

**LCA** (Logic Cell Array) - Jedná se o pole logických buněk. V podstatě jde o integrovaný obvod s vysokou hustotou CMOS.

**ASIC** (Application Specific Integrated Circuits) - Specifické aplikační integrované obvody.

**XPGA** (eXpanded Programmable Gate Array) - Rozšířené programovatelné hradlové pole.[10]

#### 2.3.1 Rozdíl mezi ASIC a FPGA

ASIC oproti FPGA je vyráběný na zakázku pro specifické konstrukční úkoly, dále mají rozdílnou hodnotovou nabídku, která musí být pečlivě vyhodnocena při výběru jednoho z nich. V minulosti byla vybrána technologie FPGA pro konstrukce s nižší rychlostí, složitostí a objemem, ale díky dnešní pokročilé technologii dosáhlo FPGA výpočetní rychlosti 500 MHz, a proto se používá i pro složitější konstrukce.

Díky velké logické hustotě a řadě dalších funkcí, kterými jsou například vestavěné procesory, bloky DSP, hodiny a vysokorychlostní sériové systémy, jsou FPGA bezkonkurenční volbou pro téměř jakýkoli typ návrhu.

### 2.3.2 Aplikace

Vzhledem k jejich možnosti programování jsou FPGA ideální pro mnoho různých aplikací. Společnost Xilinx má vedoucí postavení v oboru a poskytuje komplexní řešení. Skládá se ze zařízení FPGA, pokročilého softwaru a konfigurovatelných IP jader připravených k použití pro trhy a aplikace, jako jsou:

- ASIC Prototyping - rychlé a přesné modelování systému SoC a verifikaci vloženého softwaru
- Audio - Xilinx FPGA a cílené designérské platformy umožňují vyšší míru flexibility, rychlejší dobu uvádění na trh a nižší celkové jednorázové technické náklady pro širokou škálu audio, komunikačních a multimediálních aplikací.
- Automobilový průmysl - řešení asistenčních systémů pro řidiče a informačních systémů pro vozidla, které zaručují komfort a pohodlí.
- Broadcast a profesionální audio, video - rychlejší přizpůsobení změnám požadavků a prodloužení životního cyklu produktů pomocí platformy, která se zaměřuje na vysílání a řešení pro profesionální vysílací systémy.
- Spotřební elektronika - cenově výhodné řešení pro uživatele s plnohodnotnými spotřebitelskými aplikacemi nové generace, jako jsou konvergované sluchátka, digitální ploché displeje, informační zařízení, domácí sítě a obytné set-top boxy.
- Datové centrum - navržené pro servery s vysokou šířkou pásma, servery s nízkou latencí, sítě a úložné aplikace, které přinášejí vyšší hodnotu do cloud deployments.
- Vysoce výkonné výpočty a ukládání dat - řešení pro síťově připojené úložiště (NAS), síť Storage Area Network (SAN), servery a paměťové zařízení.
- Průmyslová výroba - designérské platformy pro průmyslové, vědecké a lékařské technologie (ISM) umožňující vyšší míru flexibility a nižší celkové jednorázové technické náklady. Využití je pro širokou škálu aplikací, jako jsou průmyslové zobrazovací a sledovací systémy, průmyslové automatizace a lékařské zobrazovací zařízení.
- Zabezpečení - Xilinx nabízí řešení, která vyhovují vyvíjejícím se potřebám bezpečnostních aplikací, od řízení přístupu až po bezpečnostní systémy.
- Zpracování videa a obrazu - designérské platformy umožňující vyšší míru flexibility, rychlejší dobu uvádění na trh a nižší celkové jednorázové technické náklady pro širokou škálu videí a zobrazovacích aplikací.



- Drátová komunikace - kompletní řešení pro zpracování reprogramovatelných sítí Line-card Packet Processing, Framer/MAC, sériové backplanes a další.
- Bezdrátová komunikace - RF, základní pásmo, konektivita, dopravní a síťová řešení pro bezdrátová zařízení.
- Lékařské - **pro diagnostické, monitorovací a terapeutické aplikace na požadavky pro zpracování, zobrazení a I/O rozhraní.[10]**

### 3 Způsoby návrhu logiky FPGA

Způsobů, jak navrhnout vnitřní logiku FPGA je několik, nejpoužívanějšími prostředky jsou Verilog, SystemVerilog, VHDL a další. Tyto prostředky slouží k popisu hardwaru (HDL - Hardware Description Language), nejedná se tedy o softwarové (SW) programování. Kromě jazyků pro popis hardware (HW) používáme také například jazyky C a C++, které slouží k ovládání navrženého HW pomocí VHDL. Pro toto SW programování můžeme využít například prostředí Eclipse.

Každý z těchto 3 jazyků má svůj vlastní styl. VHDL a Verilog implementují abstrakce na úrovni přenosu registrů (RTL) a SystemVerilog byl vyvinut, aby poskytl evoluční cestu od VHDL a Verilogu k podpoře složitosti designů SoC (system on a chip). Jazyk je kombinací HDL a jazyku ověřování hardwaru a navíc obsahuje programově orientovaný objektový přístup. Tento systém obsahuje i možnost pro testbench a ověření založené na tvrzení (assertion-based formal verification).

VHDL je silně typovaný jazyk, který na rozdíl od slabě typovaného jazyku Verilog provádí kontrolu typů a spolehlivě zjišťuje jejich chybné použití (chyby v datovém typu) za běhu programu. VHDL je více popisný, potřebuje více řádků kódů, ale jeho čitelnost je tím lepší než u jazyka Verilog. Verilog je stručnější jazyk a jeho syntaxe je podobná jazyku C.

Pro tuto práci byl zvolen jazyk VHDL, proto je více popsán v následující podkapitole.[12],[13]

#### 3.1 VHDL

Jedná se o programovací jazyk sloužící k popisu hardwarových součástí, který dokáže nasimulovat digitální integrované obvody před tím, než se dostanou do fáze výroby. Používá se i pro vytváření analogových obvodů.

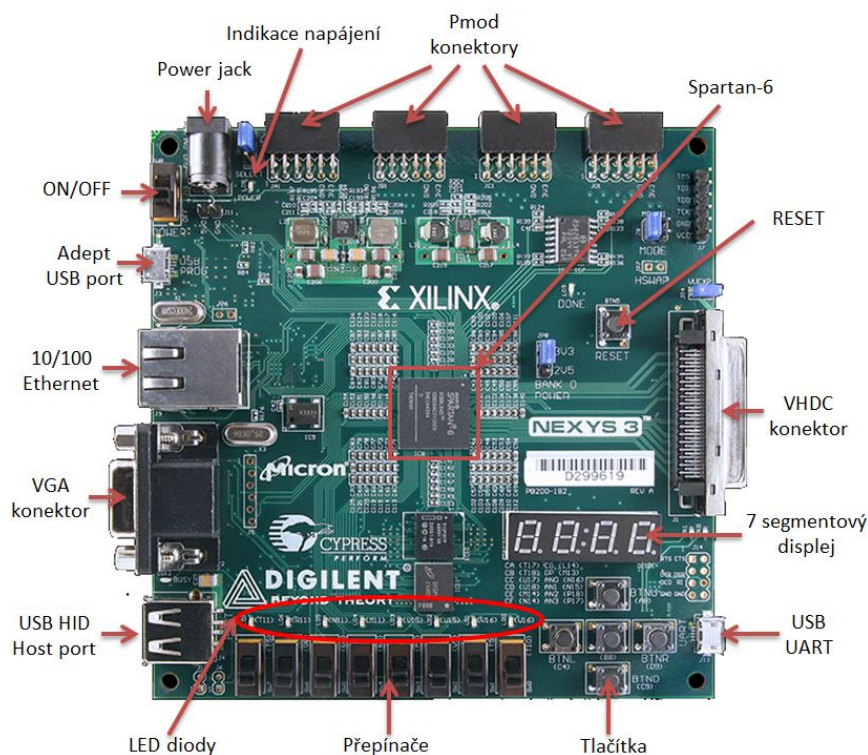
Tento programovací jazyk má kořeny v jazyce Pascal a původně byl vytvořen pro simulaci hardwaru, až později se začal využívat pro popis reálných obvodů. Výhodou je hardwarová nezávislost, tzn. napíšeme kód ve VHDL a nasimulujeme jej, aniž bychom věděli, jaký hardware bude použit. Jazyk umožňuje popsat obvody strukturálně (popsáním zapojení) nebo behaviorálně (algoritmem, jak bude daný obvod fungovat). Není case-sensitive, tudíž nezáleží na velikosti písmen.[12],[13]

## 4 Vývojová deska Nexys 3

Jedná o vývojový kit od firmy Digilent, který slouží k návrhu aplikací a testování navržených kódů pro FPGA čip Spartan-6. V podstatě se jedná o desku plošných spojů (DPS), na které je umístěn již zmíněný čip, další periferie a externí oscilátor 100MHz. Nahrání kódu je přes specifický Digilent Adept USB port, který také poskytuje napájení. Pro návrh byla vybrána právě tato vývojová deska z důvodu dostatečného výpočetního výkonu a také protože je dostupná na půdě katedry.[14]

### Periferie:

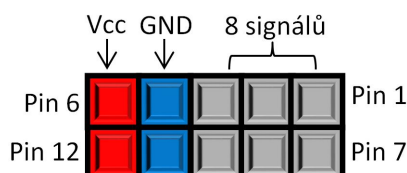
- 8 posuvných přepínačů,
- 5 tlačítek,
- 8 LED diod,
- čtyřmístný 7 segmentový displej,
- RESET tlačítko – slouží pro resetování čipu,
- USB HID Host port – pro připojení myši nebo klávesnice,
- USB-UART,
- Vysokorychlostní 40pinový konektor VHDC,
- Čtyři 12pinové Pmod konektory,
- 8bitový VGA konektor,
- 10/100 Ethernet,
- Power jack – napájení.[14]



Obrázek 10: Vývojová deska Nexys 3

## 5 Rozhraní Pmod

Rozhraní Digilent Pmod se používá pro připojení periferních modulů s nízkým kmitočtem a nízkým počtem I/O pinů na hostitelské řídicí desce (HŘD), jako je například NEXYS 3. Jsou verze 6-pinové a 12-pinové. 6-pinové verze obsahují čtyři digitální I/O signálové piny, jeden napájecí pin a jeden uzemňovací pin. Verze s 12-piny poskytují osm signálových pinů I/O, dva napájecí piny a dva zemní piny. Hostitelský port je však vždy 12-pinový viz obrázek 11.

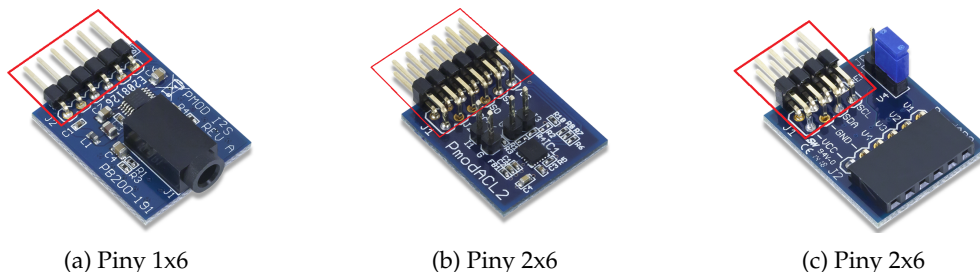


Obrázek 11: Pmod hostitelský konektor[15]

Obecně platí, že moduly Pmod se mohou připojit přímo ke konektorům HŘD nebo mohou být připojeny k řídicí desce prostřednictvím šesti nebo dvanácti-pinových kabelů. Dva šestipólové periferní moduly mohou být připojeny k jednomu dvanáctipólovému hostitelskému konektoru buď přes 12-pinový nebo dvojitý šestipólový rozdělovací kabel. Dále může být ke dvěma 6-pinovým hostitelským konektorům připojen 12-pinový modul přes stejný 12-pinový nebo dvojitý šestipólový rozdělovací kabel.

Pmodové periferní moduly jsou napájeny HŘD přes napájecí a zemní kolíky. Rozhraní Pmod není určeno pro vysokofrekvenční provoz, ale pomocí konektorů RJ45 a kabelu s krouceným párem Ethernet byly spolehlivě odesílány signály na 24 MHz na vzdálenosti až 4 metry.

Pmod má i 8-pinovou variantu s rozhraním I2C. V některých případech může být modul I2C připojen přímo do konektoru Pmod na systémové desce, obecně však bude připojen pomocí MTE kabelů. Rozhraní Pmod I2C poskytuje dva I2C signály, SDA, SCL a napájení s uzemněním. Na obrázcích níže (Obrázek 12) jdou vidět různé pinové provedení rozhraní Pmod.[16]



Obrázek 12: Rozhraní Pmod[17], [18], [19]

## 5.1 Modul PmodACL2

Jedná se o modul od firmy Digilent, na kterém je umístěn s tříosým MEMS akcelerometr ADXL362 a další součásti, které zajišťují napájení a zpracování dat. Jeho rozlišení je až 12 bitů pro každou osu akcelerace. Díky zpětné vazbě sledování neaktivity můžeme sledovat různou řadu hlášení z více zdrojů přerušení. Z toho plyne, že můžeme aktivovat režim spánku a probuzení, čímž dosáhneme vyšší úspory energie.

Tento modul také nabízí detekci volného pádu. Připojení je přes Pmod konektor a komunikace je pomocí SPI.[19],[20]

### Shrnutí:

- 3-osý MEMS akcelerometr,
- až 12 bitů rozlišení na jednu osu,
- volitelný rozsah  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  s rozlišením 1 mg/LSB při rozsahu  $\pm 2g$ ,
- Pmod konektor s komunikací přes SPI,
- monitorování aktivity/nečinnosti,
- nízká spotřeba proudu (méně než 2  $\mu A$  při 100 Hz),
- detekce volného pádu,
- malá velikost desky plošných spojů (2,5 cm  $\times$  2,0 cm),
- externí napájení v rozmezí 1,6 V a 3,5 V.[17, 18]

Tabulka 7: Popisy pinů konektoru J1[20]

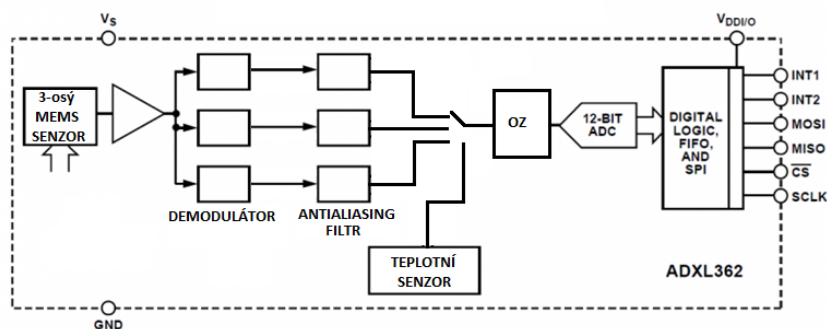
Konektor J1					
Pin	Signál	Popis	Pin	Signál	Popis
1	CS	Zvolení čipu	7	INT2	Přerušení 2
2	MOSI	Master Out Slave In	8	INT1	Přerušení 1
3	MIDO	Master In Slave Out	9	NC	Nepřipojeno
4	SCLK	Sériové hodiny	10	NC	Nepřipojeno
5	GND	Zemní svorka	11	GND	Zemní svorka
6	VCC	Napájecí napětí	12	VCC	Napájecí napětí

Tabulka 8: Popisy pinů konektoru J2[20]

Konektor J2		
Pin	Signál	Popis
1	INT1	Přerušení 1
2	GND	Zemní svorka

Tabulka 9: Popisy pinů konektoru J3[20]

Konektor J3		
Pin	Signál	Popis
1	INT2	Přerušení 2
2	GND	Zemní svorka



Obrázek 13: Funkční diagram akcelerometru ADXL362[21]

### Popis funkčního diagramu

Naměřená data se nejprve demodulují v demodulátoru, dále prochází přes antialiasingový filtr, kde dojde k ošetření podvzorkování. Následuje operační zesilovač, ve kterém dojde k zesílení a následně k navzorkování v ADC (A/D) převodníku. Hodnoty jsou pak uloženy ve FIFO paměti, kde jsou uloženy konfigurační registry. Modul disponuje teplotním senzorem a externími hodinami o kmitočtu 51,2 kHz.[21]

## 5.2 Modul PmodSTEP pro řízení krokových motorů

PmodSTEP komunikuje s HŘD prostřednictvím protokolu GPIO. Tento modul může řídit 4-pinový a současně 6-pinový krokový motor. Krokové motory pracují střídavě napájením cívek na různé polarity, které indukují krokový motor k otáčení. Detailnější vysvětlení viz kapitola 1.

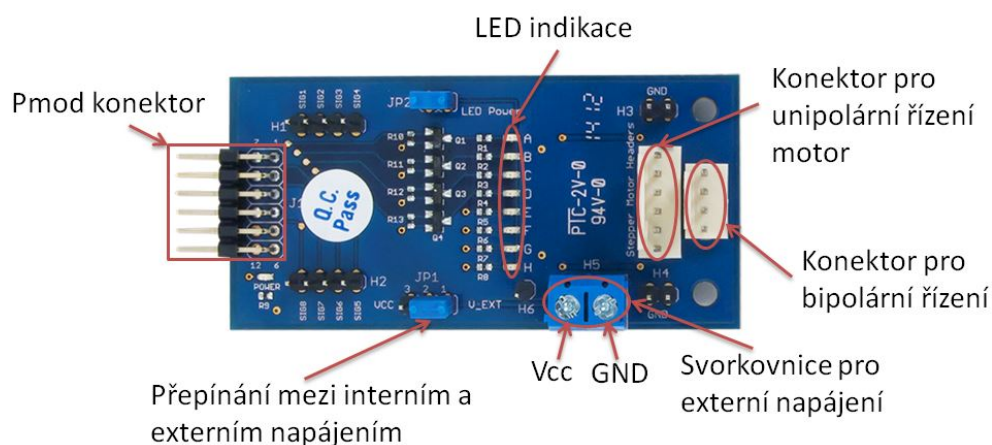
4-pinové krokové motory pracují pouze v bipolárním uspořádání a vyžadují, aby dva vstupy na každé elektromagnetické cívice byly přivedeny na správné logické napěťové úrovně tak, aby indukovaly proudový tok ve správném směru.

6-pinové krokové motory mohou být orientovány buď pro bipolární nebo unipolární konfiguraci.

PmodSTEP využívá čtyřkanalový ovladač ST L293DD k řízení krokových motorů při vyšších proudech, než systémová deska obvykle poskytuje z jejich logických výstupů. Pro snadné testování a sledování šíření signálů jsou k dispozici externí testovací body a LED diody.[22]

### Funkce modulu:

- může pohánět 2 motory současně,
- LED indikace buzení výstupů,
- propojka pro externí napájení,
- malá velikost DPS 2,8" × 1,3" (7,1 cm × 3,3 cm),
- 2x 6-pinový konektor Pmod s rozhraním GPIO.[22]



Obrázek 14: Modul PmodSTEP[22]

Tabulka 10: Rozdělení pinů modulu PmodSTEP[22]

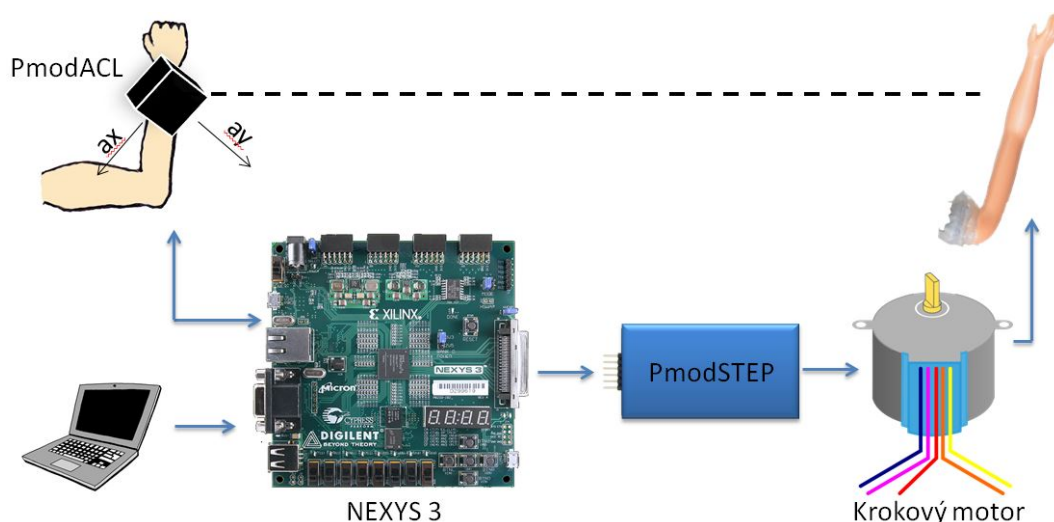
Pin	Signál	Popis
1	SIG1	Signál 1
2	SIG2	Signál 2
3	SIG3	Signál 3
4	SIG4	Signál 4
5	GND	Zemní svorka
6	VCC	Napájecí napětí
7	SIG5	Signál 5/Výstup 1 pro krokový motor
8	SIG6	Signál 6/Výstup 2 pro krokový motor
9	SIG7	Signál 7/Výstup 3 pro krokový motor
10	SIG8	Signál 8/Výstup 4 pro krokový motor
11	GND	Zemní svorka
12	VCC	Napájecí napětí

Jakékoliv externí napájení musí být mezi 4,5–36 V, doporučená úroveň napětí je 5 V.[22]



## 6 Návrh koncepce výukového kybernetického modelu s krokovými motory a měřením polohy

Návrh koncepce modelu je takový, že se k vytvořenému modelu ruky o jednom kloubu připojí krokový motor PM42L-048 od firmy Minebea. Jako model ruky bude využita paže od panenky, která bude připevněna ke hřídeli krokového motoru. Motor bude ovládán modulem PmodSTEP, který bude připojen na desku NEXYS 3. Ovládání pohybu krokového motoru bude pomocí dvou akcelerometrů, které budou připnuté na paži uživatele. Jeden z akcelerometrů bude připnut na předloktí (antebrachium) a druhý bude umístěn na dvojhlavém svalu pažním (musculus biceps brachii). Použití dvou akcelerometrů je z důvodu reakce krokového motoru pouze na přitažení a odtažení (flexi a extenzi) paže.



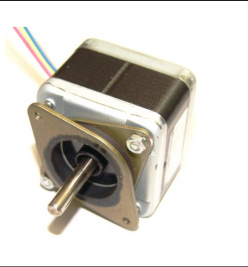
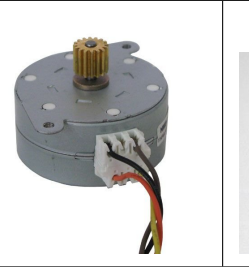

Obrázek 15: Blokové schéma návrhu výukového modelu[23],[24],[25]

Deska NEXYS 3 je napájena zdrojem stejnosměrného napětí (5 V) z USB. Vzhledem k tomu, že použitý krokový motor potřebuje větší napájení než je schopen dodat modul PmodSTEP, je potřeba přivést do modulu externí napájení. Úroveň napětí musí být v povoleném rozsahu modulu, jinak by došlo k jeho porušení. Stejně tak je napájení omezeno použitým krokovým motorem.

### 6.1 Výběr vhodného krokového motoru

Výběr krokového motoru byl omezen dostupností a prostředky katedry, tudíž lze na trhu najít i lepší model. Práce slouží jako podklad pro možné laboratorní úlohy, proto byly vybrány motory dostupné na půdě katedry. Požadavky jsou malé rozměry, napájení malým napětím pro laboratorní účely a dostatečný počet kroků. V následující Tabulce 11 jsou vidět specifikace vybraných motorů, které byly porovnávány.

Tabulka 11: Porovnání různých dostupných krokových motorů[26], [27], [28], [29], [30]

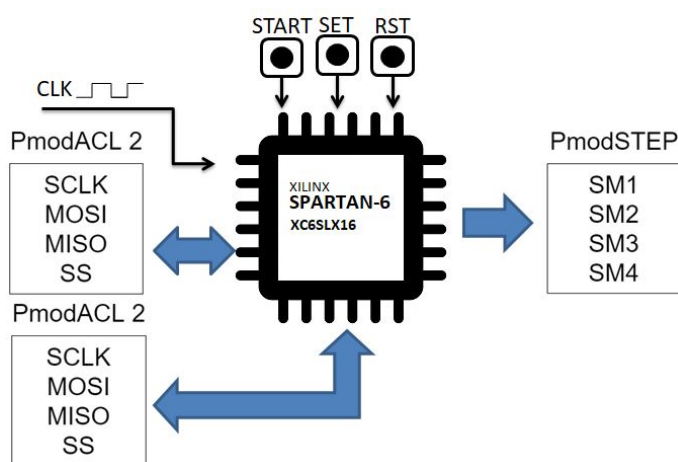
			
<b>Firma</b>	Minebea	Minebea	Minebea
<b>Série</b>	17PM-K	PM42L	34PM-C
<b>Model</b>	17PM-K502-G2ST	PM42L-048	34PM-C101
<b>Typ</b>	Hybridní	PM	Hybridní
<b>Typ buzení</b>	Bipolární	Bipolární	Unipolární
<b>Počet kroků na otočku</b>	200	48	200
<b>Úhel kroku</b>	1,8°	7,5°	1,8°
<b>Rozměry motoru</b>	42 × 42 × 33 mm	49,5 × 22,2 mm	85,7 × 93,7 mm
<b>Délka hřídele</b>	50 mm	12 mm	30,2 mm
<b>Napájecí napětí</b>	12-36 V	24 V	3 V
<b>Odebíraný proud</b>	1,2 A	0,6 A	5 A

Motor PM42-L048 má optimální odebíraný proud, ale nedostačující úhel kroku. Model 34PM-C101 má dostačující krokový úhel, zato několikanásobně přesahující maximální proud modulu PmodSTEP a velké rozměry. Z dostupných motorů byl vybrán model 17PM-K502-G2ST z důvodu krokového úhlu a odebíraného proudu, sice také přesahuje dovolený limit pro modul PmodSTEP, ale s proudovým omezením zdroje je pro tuto úlohu nejvhodnější.

Za zmínku stojí napájecí napětí, kdy se motor otáčí už při 3,6 V, s narůstající hodnotou napětí roste i síla motoru. Tato informace je důležitá vzhledem k mému použití motoru, neboť kdyby se robotická paže zatížila závažím mohlo by se stát, že by motor „proklouzl“, to by způsobilo následnou odchylku mezi robotickou paží a vyhodnocováním polohy akcelerometrem. Dalším faktem je, že motor není využit naplno, což je způsobeno proudovým omezením modulu PmodSTEP.

## 7 Návrh obvodové logiky laboratorní úlohy s modulem PmodSTEP pro vývojovou desku s FPGA

Návrh logiky je označen jako entita `kyberneticky_model`. Tahle entita je nejvýše položenou entitou v hierarchii návrhu. Diagram připojení periferie k FPGA je zobrazen na následujícím obrázku (Obrázek 16). Ve schématu jsou zobrazeny vlevo (vstupní) a vpravo (výstupní) signály. Signál CLK je vstupní hodinový signál o frekvenci 100 MHz, který je generovaný krystalovým oscilátorem na desce Nexys 3. Jsou zde připojena tři tlačítka START, SET a RST. START slouží pro zapnutí pohybu krokového motoru, tlačítko RST sloužící jako reset a SET pro nastavení výchozího bodu od kterého bude umělá ruka zrcadlit uživatelskou paži. Pro sériovou linku slouží signály SDI, SDO, SCLK a SS.



Obrázek 16: Diagram připojení periferie k FPGA[31]

Kompletní návrh obvodové techniky je rozdělen do několika dalších modulů (komponent). Ty jsou v následujících kapitolách podrobněji popsány.

### 7.1 Komponenty návrhu

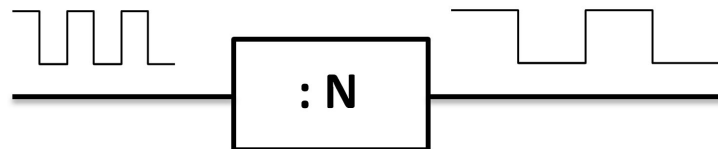
Entita `kyberneticky_model` se rozděluje do sedmi komponent. Každá komponenta splňuje jinou funkci a komponenty jsou navzájem různě propojené. Celá entita je řízená hodinovým signálem CLK, který časuje obvody podle svého výchozího kmitočtu (100 MHz u NESYS 3). Komponenta `Signal_5Hz` jej dělí na signál o kmitočtu 5 Hz, který se dále využívá při komunikaci s akcelerometrem v komponentě `ACL_SPI`, kde se využívá pro odesílání a přijímání dat. Příchozí data jsou v této komponentě rozdělovány na 3 datové vektory x, y, z. Tato data jsou ve tvaru dvojkového doplňku a jejich šířka je 12 bitů. Následuje komponenta `data_processing_component`, ve které se omezují kmitočty paže a následně se odstraňuje snímaný šum pomocí klouzavého průměru. Další komponentou je IP jádro Cordic vypočítávající úhel natočení akcelerometru. Finální komponentou je `data_evaluation`, kde se vyhodnocuje, zda se má provést krok a jakým směrem. V příloze se nachází RTL schéma vnitřních komponent.

Tabulka 12: Přehled I/O pinů entity *kyberneticky\_model*

Název	I/O pin	Směr	Popis
CLK	V10	IN	hodinový signál (100 MHz)
btnd	C9	IN	tlačítko RESET
btns	B8	IN	tlačítko SET
btneu	A8	IN	spínač STARTSTEP
LED 0	U16	OUT	LED indikace
JA 0	T12	OUT	SS = slave select první akcelerometr
JA 2	V12	OUT	SDO = slave data out první akcelerometr
JA 3	N10	IN	SDI = slave data in první akcelerometr
JA 4	P11	OUT	SCLK = slave CLK první akcelerometr
JB 0	K2	OUT	SS1 = slave select druhý akcelerometr
JB 1	K1	OUT	SDO1 = slave data out druhý akcelerometr
JB 2	L4	IN	SDI1 = slave data in druhý akcelerometr
JB 3	L3	OUT	SCLK1 = slave CLK druhý akcelerometr
JD 4	D12	OUT	výstup na Pmod konektor JD
JD 5	C12	OUT	výstup na Pmod konektor JD
JD 6	F12	OUT	výstup na Pmod konektor JD
JD 7	E12	OUT	výstup na Pmod konektor JD

## 7.2 Komponenta Signal\_START

Tato komponenta slouží pro generování start signálu, který zahájí komunikaci s akcelerometrem. Tento signál vznikne dělením CLK, který má v tomto případě kmitočet 100 MHz.



Obrázek 17: Blokové schéma děličky hodinového signálu

Ve VHDL je tvořen pomocí procesu, který lze vidět na obrázku 18. Do procesu vstupují signály CLK a RST (reset). Když dojde ke změně stavu některého z těchto signálů, proces se provede. Dále v procesu můžeme vidět signál count a CLKOUT. Signál count čítá náběžné hrany CLK a CLKOUT je výstupní generovaný signál. Celý proces je kontrolován několika podmínkami. První podmínka hlídá, zda nebyl vyžádán reset (zmáčknutí tlačítka). Když dojde ke změně z log. 0 na log. 1, pak všechny výchozí signály budou nastaveny na hodnotu 0. Následující podmínka kontroluje náběžnou hranu CLK. Pokud přijde náběžná hrana a hodnota count nedosáhla maxima (EndVal), tak se k signálu count přičte jednička. Tohle se opakuje, dokud nedosáhne hodnota count maxima, poté se na CLKOUT objeví log. 1 a čítač se vynuluje. Ma-

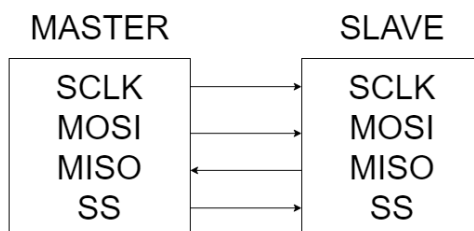
ximální hodnota je nastavena na 10 000 000. Když se celý proces zopakuje, tak se na CLKOUT objeví log 0. Tím dojde ke generování signálu se skoro libovolným kmitočtem.

```
process(CLK, RST) begin
    if(RST = '1') then
        CLKOUT <= '0';
        count <= X"000000";
    elsif rising_edge(CLK) then
        if(count = EndVal) then
            CLKOUT <= NOT CLKOUT;
            count <= X"000000";
        else
            count <= count + '1';
        end if;
    end if;
end process;
```

Obrázek 18: Část kódu znázorňující děličku hodinového signálu

### 7.3 Komunikace s akcelerometrem

Komunikace s akcelerometrem je zajištěna přes Serial Peripheral Interface dále jen SPI. Jedná se o sériové rozhraní, které zajišťuje komunikaci jednoho čipu s jedním nebo více čipy. Používány jsou sice 2 akcelerometry, ale každý má své rozhraní, takže se jedná o komunikaci jednoho čipu s druhým. Značení je slave pro akcelerometr a master pro navrženou logiku na HŘD. Celá komunikace probíhá pomocí 4 vodičů: SCLK – serial clock, MOSI – master out slave in, MISO – master in slave out a SS – slave select.



Obrázek 19: Komunikace přes SPI

Datový formát pro komunikaci vypadá následovně

< /SSdown > < commandbyte > < addressbyte > < databyte > < /SSup >

Zařízení master zahájí komunikaci nastavením výstupu slave select do logické 0. Tím se vybere některé zařízení, se kterým bude master komunikovat (v rámci této práce se jedná pouze o jedno zařízení). SCLK je mastrem generovaný signál, který určuje rychlost přenosu jednotlivých bitů. Každou změnou jeho hodnoty dojde k odeslání jednoho bitu. Data se přenáší vodiči MOSI a MISO. Přenos probíhá ve full-duplex, což znamená, že s hranou SCLK dojde k přenosu dat z mustru do slave a naopak. Po ukončení přenosu se nastaví SS do log. 1.

Data a konfigurace se ukládají do registrů akcelerometru. Při komunikaci využívající SPI se komunikuje přes protokol, který nám určuje formu balíčků, jakou komunikuje HŘD s akcelerometrem. Na začátku zprávy se SS nastaví do log. 0 (/SS down), to zahájí komunikaci, pak následuje příkaz, jestli se bude jednat o čtení nebo zapisování, dále je adresa registru a samotná data. Proces se ukončí nastavením SS do log. 1 (/SS up).

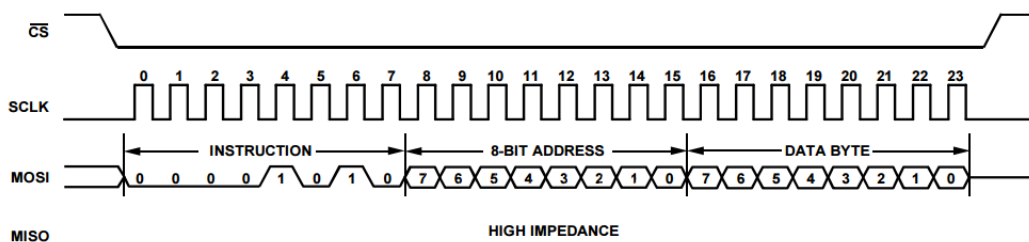
Jak je výše zmíněno, akcelerometr je nutné nejprve nakonfigurovat, což se provádí zápisem těchto hodnot do registrů akcelerometru, hodnoty a adresy jsou vypsané v následující tabulce (Tabulka 13).

Tabulka 13: Konfigurace ACL[20]

Název registru	Adresa registru	Zapsaná data
POWER_CTL	0x2D	0x0A
TRESH_ACT_L	0x20	0xFA
TRESH_ACT_H	0x21	0x00
TRESH_INACT_L	0x23	0x96
TRESH_INACT_H	0x24	0x00
TIME_INACT_L	0x25	0x1E
ACT_INACT_CTL	0x27	0x3F
INTMAP2	0x2B	0x0A

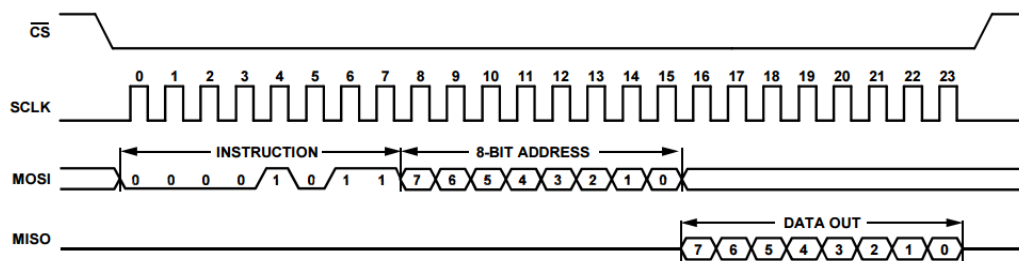
Použitý akcelerometr má plno různých funkcí a možností nastavení, které se provádí různými konfiguracemi. Akcelerometr je nastaven do režimu wake-up módu, ten detekuje pohyb při velmi nízké spotřebě. Rozsah měření je zvolen na  $\pm 2g$ .

Nejprve se nastavuje akcelerometr. Ten se nastavuje pokaždé, když přivedeme na akcelerometr napájení. Strukturu příkazu lze vidět na Obrázku 20, kde se nejprve zapíše 8 bitový příkaz pro zápis (hodnota 0x0A, 00001010), následuje 8 bitů adresy registru a nakonec 8 datových bitů.



Obrázek 20: Časový diagram zápisu do registru modulu PmodACL2[20]

Na dalším obrázku (Obrázek 21) je zobrazeno čtení z akcelerometru. Zde se místo příkazu pro zápis (0x0A) zapíše příkaz pro čtení (0x0B, 00001011), poté se odešle do senzoru 8 adresových bitů a akcelerometr odpoví po vodiči MISO 8 bitů naměřených dat.



Obrázek 21: Časový diagram čtení z registru modulu PmodACL2[20]

Ke čtení slouží registry vypsané v Tabulce 14. Po SPI lze poslat jen 8 bitové data, ale akcelerometr je 12 bitový, což znamená, že pro získání 12 bitové hodnoty zrychlení je potřeba číst vždy ze 2 registů pro každou osu. V registru končícím `_L` je uloženo horních 8 bitů [11:4] a v registru končícím `_H` jsou zbývající bity [3:0].

Tabulka 14: Datové registry modulu PmodACL2[20]

Název registru	Adresa registru
XDATA_L	0x0E
XDATA_H	0x0F
YDATA_L	0x10
YDATA_H	0x11
ZDATA_L	0x12
ZDATA_H	0x13

## 7.4 Komponenta ACL\_SPI

ACL\_SPI komponenta zajišťuje veškerý přenos dat mezi modulem PmodACL 2 a deskou Nexys 3. Tyto komponenty jsou v návrhu dvě (pro každý modul akcelerometru). Samotná komponenta se rozděluje do 3 dalších komponent SPI\_control, data\_transmission, slave\_connection.

Komponenta SPI\_control je napsána jako konečný stavový automat FSM (Fine State Machine), která řídí komponenty data\_transmission a slave\_connection pomocí signálu transmit a done. Data\_transmission vytváří signál SCLK, který slouží k časování přenosu bitů při komunikaci, dále přenáší data do modulu akcelerometru. A přijaté data ukládá je do bufferu txbuffer, kde jsou data předány řídicí komponentě SPI\_control. Zde se data rozdělí do jednotlivých signálů xAxis, yAxis a zAxis. Slave\_connection vytváří signál SS, který zahajuje a ukončuje přenos dat.[20], [32]

### 7.4.1 SPI\_control

Do komponenty vstupuje signál START odesílaný komponentou Signal\_5Hz, který řídí začátek přenosu dat. Nejprve se provádí konfigurace akcelerometru a následně se z něj čtou naměřená



data xAxis, yAxis a zAxis. Stavový automat může mít několik stavů idle, configure, finished, break, holding, transmitting a receiving. První stav je stav idle, ve kterém se řeší zda je akcelerometr nakonfigurován, tento stav můžeme resetovat tlačítkem btnd, který se nachází na desce Nexys 3. V případě, že komunikace proběhla úspěšně, přejde automat do stavu transmitting, v případě, že konfigurace neproběhla správně přejde do stavu configure a provede se konfigurace. Po dokončení konfigurace přejde automat do stavu finished.

Pokud je automat ve stavu idle a konfigurace už proběhla, přejde do stavu transmitting a provede se čtení z datových registrů os. Po ukončení přenosu všech dat přejde automat do stavu break, kde se hlídá zda bylo čtení úspěšné, pokud tomu tak není, tak se změní stav na transmitting nebo configure. Po úspěšném čtení přejde ze stavu break na holding a znovu se zahájí nové čtení dat. V příloze se nachází RTL schéma komponenty ACL\_SPI.

#### **7.4.2 data\_transmission**

Komponenta data\_transmission slouží k odesílání a přijímání dat skrze SPI a také generuje signál SCLK, který časuje komunikaci s modulem akcelerometru. Odesílaná data jsou přenášena skrze výstup SDO a přijímána skrz vstup SDI. Když přijde od SPI\_control signál transmit, přesunou se odesílané data do posuvného registru Txbuffer, odtud se postupně posílají do PmodACL 2. Naopak přijaté data se ukládají do bufferu Rxbuffer. Ukončením přenosu se data odešlou do SPI\_control.

#### **7.4.3 slave\_connection**

Tato komponenta kontroluje stav signálu SS. V případě, že je tento signál v logické 1, komunikace neprobíhá. Pokud chceme začít komunikaci s akcelerometrem musíme tento signál nastavit do logické 0, tato změna je vyžádána komponentou SPI\_control prostřednictvím signálu transmit.

### **7.5 Komponenta data\_processing\_component**

Následuje komponenta data\_processing\_component ve které jsou implementovány další čtyři komponenty y\_avg, z\_avg y1\_avg a z1\_avg. V samotné komponentě data\_processing\_component jsou ošetřeny kmity paže. Pokud by se uživatel rozhodl rychle měnit úhel paže, tak by mohlo při větší zátěži dojít k „proklouznutí“ zubů krokového motoru, což by znamenalo ztrátu aktuální pozice. Jedná se o jednoduchý proces, jehož výstup je signál CEO, který je přivedený na clock enable (povolení hodin) komponenty Cordic. Dále je zde dělička hodinového signálu pro řízení výpočtu v následujících komponentách.

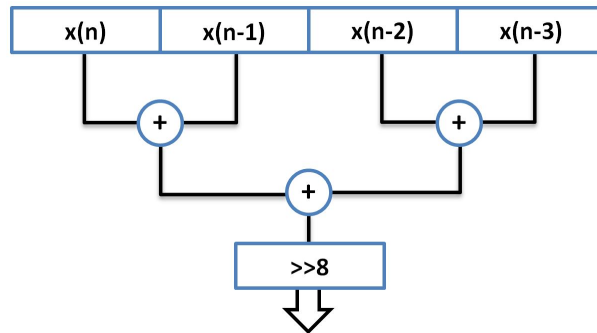
```

-- omezení pro kmity paže
if (zAxis+yAxis) < "010100000000" then
    CEO <= '1';
else
    CEO <= '0';
end if;

```

Obrázek 22: Fragment VHDL kódu pro omezení kmitů paže

V této části se také řeší šum akcelerometru. I v případě, že je modus PmodACL2 v klidové poloze, dochází k chybnému snímání. Tato chyba se projeví šumem, což má za následek špatné vyhodnocení aktuálního úhlového natočení. Díky tomu se vyhodnotí změna a krokový motor se pootočí. Odstranění šumu je pomocí klouzavého průměru, který se počítá pro každou osu a je řešen v komponentách `y_avg`, `z_avg`, `y1_avg` a `z1_avg` (pro jeden a druhý modul PmodACL2).



Obrázek 23: Diagram principu klouzavého průměru[33]

### Klouzavý průměr

Jedná se o průměr, kde se sečtou dvě naměřené hodnoty a jejich součet se sečte se součtem dalších dvou naměřených hodnot. Počet hodnot, ze kterých se vypočítává průměr, ovlivňuje míru „vyhlazení“. Poté, co máme čísla sečtená je potřeba je vydělit číslem  $n$ . Pokud  $n = 2^i$ , dělení se řeší tak, že se nepočítá se spodními  $i$  bity výsledku.[34]

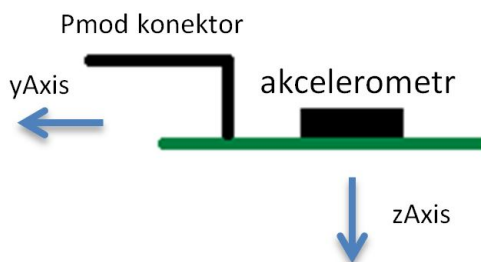
## 7.6 Komponenta arctg

Jedná se o IP jádro, které slouží pro výpočet goniometrických funkcí. V tabulce 15 můžeme vidět I/O signály. Komponenta je v návrhu dvakrát, pro výpočet úhel natočení obou akcelerometrů vůči zemi. Výpočet uhlu je pomocí funkce  $\arctg \frac{y}{x}$ , kde v mém případě jsou vstupní argumenty osa  $z$  (`x_in`) a  $y$  (`y_in`), tato volba je z konstrukčních důvodů (umístění v krabici a následné připnutí na ruku).

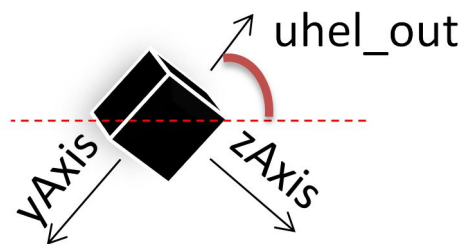
Tabulka 15: Vstupy a výstupy komponenty arctg

Název	I/O	Popis
x_in	IN	argument x
y_in	IN	argument y
phase_out	OUT	výsledný úhel
CE	IN	povolení hodin
CLK	IN	hodinový signál

Na Obrázku 24 můžeme vidět, že osa z je kolmá na DPS modulu PmodACL 2 a osa y je po směru konektoru.



Obrázek 24: Pohled z boku na modul PmodACL2



Obrázek 25: Znázorněn výpočet úhlu

Do komponenty vstupuje také signál CE, který povoluje výpočet. Tento signál je zde proto, aby nedocházelo k chybným výpočtům při velmi rychlých pohybech ruky, kdy jsou osy ovlivněny odstředivou silou.

Vstupní signál je ve formě dvojkového doplňku s pevnou desetinnou čárkou ležící za 2 bity (1Q10 formát) a výsledný úhel je ve formě desetinného čísla s pevnou desetinnou čárkou za 3 bity (2Q9 formát). Rozmezí hodnot a tvar je znázorněn v tabulkách níže. Pro vysvětlení počítáme se vstupem ve formě 1Q7(Fix9.7) a výstupem 2Q6(Fix9.6).

Tabulka 16: 1Q7 formát dat[35]

	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+1	0	1	0	0	0	0	0	0	0
-1	1	1	0	0	0	0	0	0	0

Tabulka 17: 2Q6 formát dat[35]

	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+Pi	0	1	1	0	0	1	0	0	1
-Pi	1	0	0	1	1	0	1	1	1

## 7.7 Komponenta data\_evaluation

Tato komponenta je finální komponentou této práce. V komponentě dochází k vyhodnocení, zda se má provést krok a kterým směrem. V následující tabulce je seznam I/O signálů této komponenty.

Tabulka 18: Přehled I/O signálů komponenty data\_evaluation

Název	I/O	Popis
CLK	IN	hodinový signál (100 MHz)
RST	IN	tlačítko reset
SET	IN	tlačítko set (nastavení)
STARTSTEP	IN	tlačítko start
UHEL_OUT	IN	úhel natočení prvního modulu PmodACL2
UHEL_OUT1	IN	úhel natočení druhého modulu PmodACL2
START	IN	zahájení přenosu dat z akcelerometru
LED	OUT	LED indikace
JA	OUT	výstup na modul PmodSTEP

Komponenta je obsáhlá a rozděluje se do několika částí. Tyto části mají odlišné funkce a jsou na sebe různě napojené.

Části jsou:

- konverze,
- proces tlačítko,
- dělička CLK,
- rozdíl,
- proces krokování,
- vyhodnocení kroku.

Pro vyhodnocení polohy bylo potřeba konvertovat vstupní signál uhel\_out z STD.LOGIC\_VECTOR na integer. Proto byl vytvořen nový datový typ dojdopln, který nabývá hodnot od -2048 do 2047. Následně se do signálu uhel\_conv uloží konvertovaná hodnota.

subtype dojdopln is integer range -2048 to 2047;

signal uhel\_conv : dojdopln; – pro úhel natočení prvního akcelerometru

signal uhel\_conv1 : dojdopln; – pro úhel natočení druhého akcelerometru

uhel\_conv <= conv\_integer (uhel\_out);

uhel\_conv1 <= conv\_integer (uhel\_out1);

Signál STARTSTEP vede z tlačítka btu, které je zapojené jako spínač, tzn. při jednom stisknutí se na tomto signálu objeví logická 1 a hodnota je tam dokud se tlačítko znovu nezmáčkne. Nejprve se zkontroluje náběžná hrana signálu CLK, poté se kontroluje zda je tlačítko stisklé. Pokud ano, tak se do startstep\_old uloží hodnota 1. Když dojde k uvolnění tlačítka, tak se invertuje signál TL a do startstep\_old se uloží logická 0.

```
tlacitko: process (STARTSTEP, CLK)
begin
  if rising_edge(CLK) then
    if STARTSTEP = '1' then
      startstep_old <= '1';
      if startstep_old = '0' then
        TL <= NOT TL;
      end if;
    else
      startstep_old <= '0';
    end if;
  end if;
end process;
```

Obrázek 26: Fragment VHDL kódu pro tlačítko

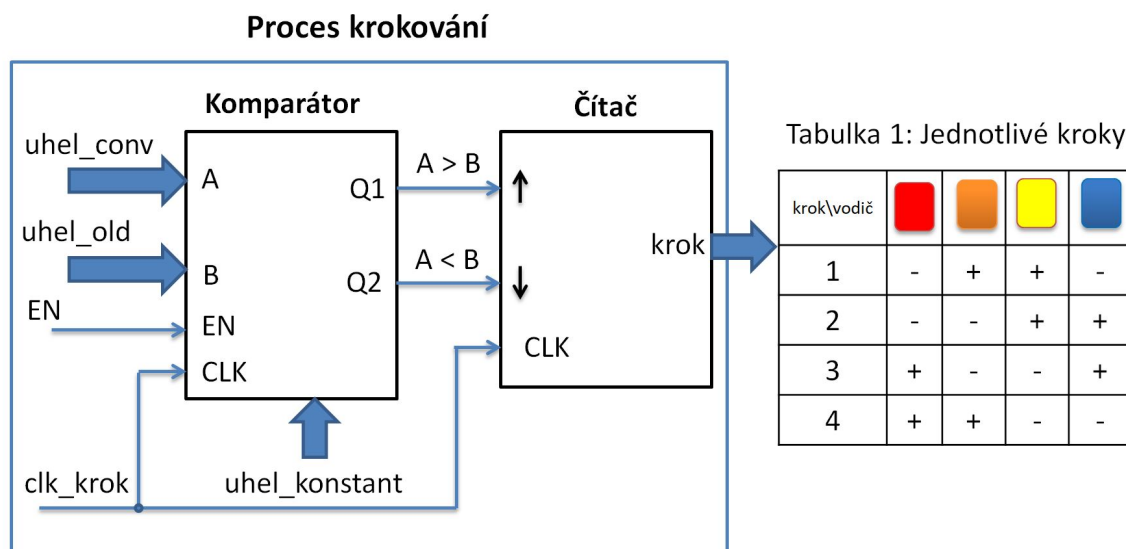
Rychlost krokování je řízena děličkou, která dělí hodinový signál CLK o hodnotě 100 MHz na hodnotu 66,6 Hz. Dělička je zde, protože musíme přizpůsobit rychlost krokování fyzickým možnostem krokového motoru. Kdybychom použili signál o vysoké frekvenci, tak by došlo ke ztrátě kroků. Princip děličky je vysvětlen v kapitole Komponenta Signal.START.

```
delicka: process (CLK)
begin
  if Rising_Edge (CLK) then
    if count = 2500000 then
      count <= (others => '0');
      clk_krok <= not clk_krok;
    else
      count <= count + 1;
    end if;
  end if;
end process delicka;
```

Obrázek 27: Část VHDL kódu pro děličku hodinového signálu

### Proces rozdíl

Jedná se o velmi jednoduchý proces, ve kterém se vyhodnocuje rozdíl úhlů obou akcelerometrů. V případě, že hodnota  $uhel\_conv > uhel\_conv1$  (hodnota úhlu prvního akcelerometru umístěného na zápěstí je větší než hodnota druhého akcelerometru), do signálu EN se zapíše logická hodnota 1 v opačném případě se do tohoto signálu zapíše logická 0.



Obrázek 28: Diagram procesu krokování

### Princip procesu krokování

Celý proces se skládá z komparátoru a čítače a je časován výstupem `clk.krok` z děličky hodinového signálu. Je zde signál `EN` (enable = povolení), který v případě logické hodnoty 1, povolí porovnání. Dále do komparátoru vstupuje konvertovaný úhel (`uhel_conv`), minulá hodnota úhlu (`uhel_old`) a konstanta (`uhel_konstant`). Porovnává se zde zda je `uhel_conv` rozdílný oprati `uhel_old`, ale to by nestačilo. Způsobilo by to vyhodnocení rozdílů a krokový motor by neustále dělal krok doprava a doleva kolem jedné hodnoty, proto je zde přivedena i konstanta, která nabývá hodnoty úhlu jednoho kroku motoru.

Výsledné vyhodnocení je tedy takové, že se porovná zda je  $(uhel\_conv - uhel\_old) > uhel\_konstant$ . V případě, že je splněna podmínka se na výstupu Q1 objeví logická 1 a na výstupu Q2 logická 0. V opačném případě, kdy je `uhel_conv` menší než předcházející hodnota `uhel_old` tak se vyhodnotí zda  $(uhel\_old - uhel\_conv) > uhel\_konstant$  a pokud je podmínka splněna objeví se na Q1 logická 0 a na Q2 logická 1. Po vyhodnocení se do `uhel_old` zapíše hodnota z `uhlu_conv`.

Výstup z komparátoru je přiveden na vstup čítače, kde ovlivňuje směr čítání viz Obrázek 28

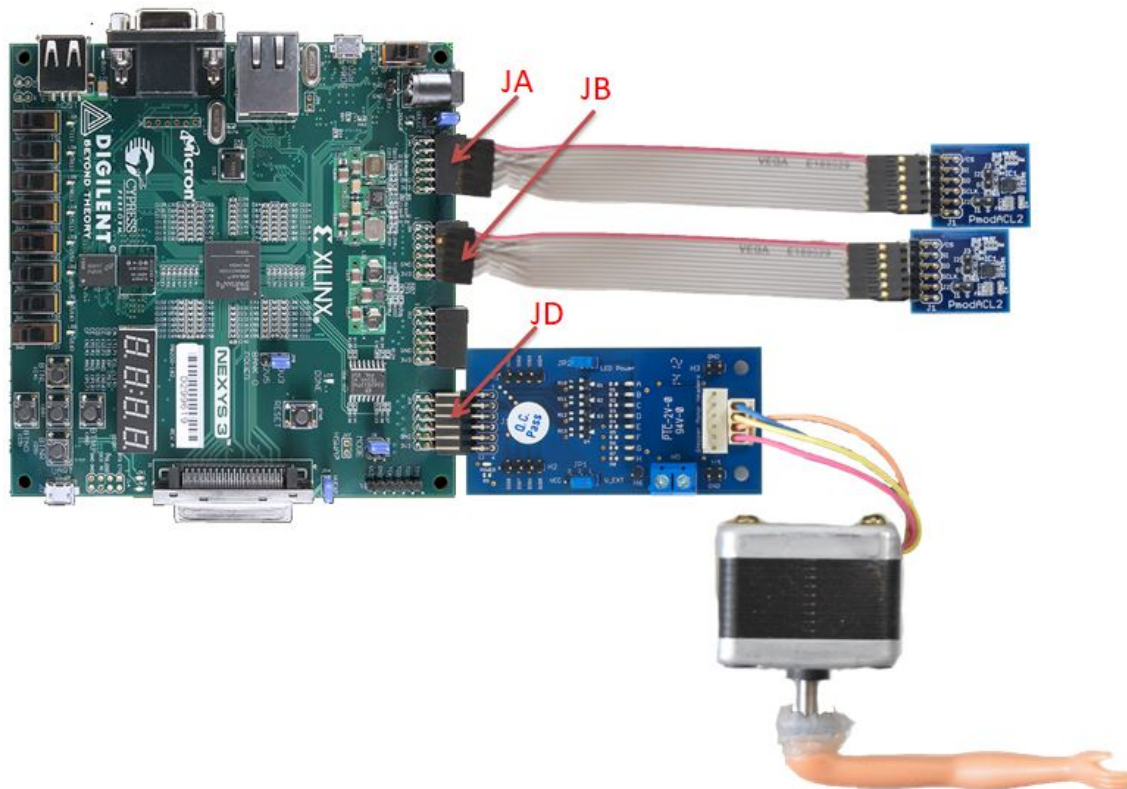
Na následujícím obrázku je výstup z čítače připojen na Pmod konektor modulu PmodSTEP.

```
with krok select JD <=
    "1100" when "00",
    "1001" when "01",
    "0011" when "10",
    "0110" when "11";
```

Obrázek 29: VHDL kód pro vyhodnocení kroků

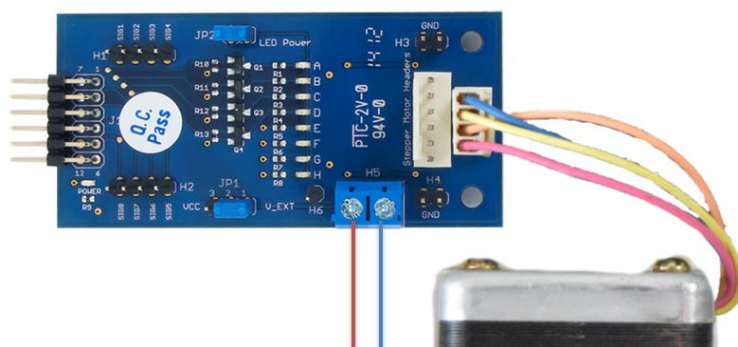
## 8 Realizace fyzikálního výukového modelu s krokovými motory

Model se povedlo úspěšně sestavit a oživit, jeho výslednou podobu se zapojenými moduly lze vidět na obrázku 30.



Obrázek 30: Celkový pohled na model

Na obrázku 31 můžeme vidět detailní zapojení motoru do modulu PmodSTEP. Důležité je pořadí vodičů, které vedou do patice na modulu. Jejich záměna je možná, ale musí se upravit VHDL, aby motor správně pracoval.

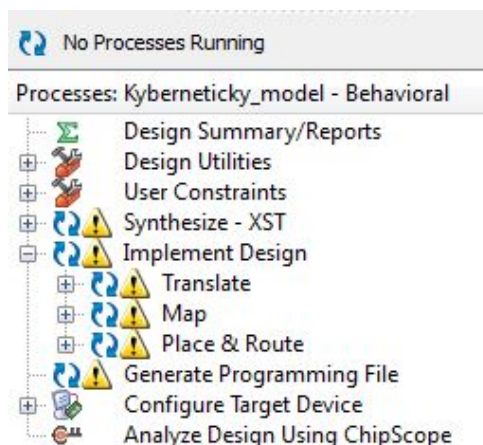


Obrázek 31: Detail zapojení PmodSTEP s krokovým motorem



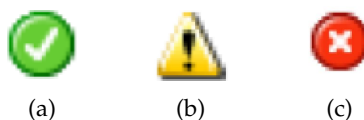
## 9 Implementace, oživení a ověření funkce navržené logiky pro FPGA

Po vytvoření všech komponent dojde k procesu implementace, ve které je zahrnuta syntéza, překlad, mapování, rozmísťování a propojování. Implementaci zahájíme spuštěním „**Implement Design**“.



Obrázek 32: Implementace

Po ukončení implementace se u každé části můžou zobrazit tři různé symboly, když byl nějaký krok úspěšný, tak se zobrazí symbol „fajfka“ (OK = v pořádku), . V případě nevhodného návrhu se zobrazí symbol vykřičník (warning = varování), toto varování neukončí implementaci, ale je vhodné si přečíst hlášení. Pokud se zobrazí symbol křížku (error = chyba), tak se ukončí aktuální krok implementace a je nutné chybu odstranit.

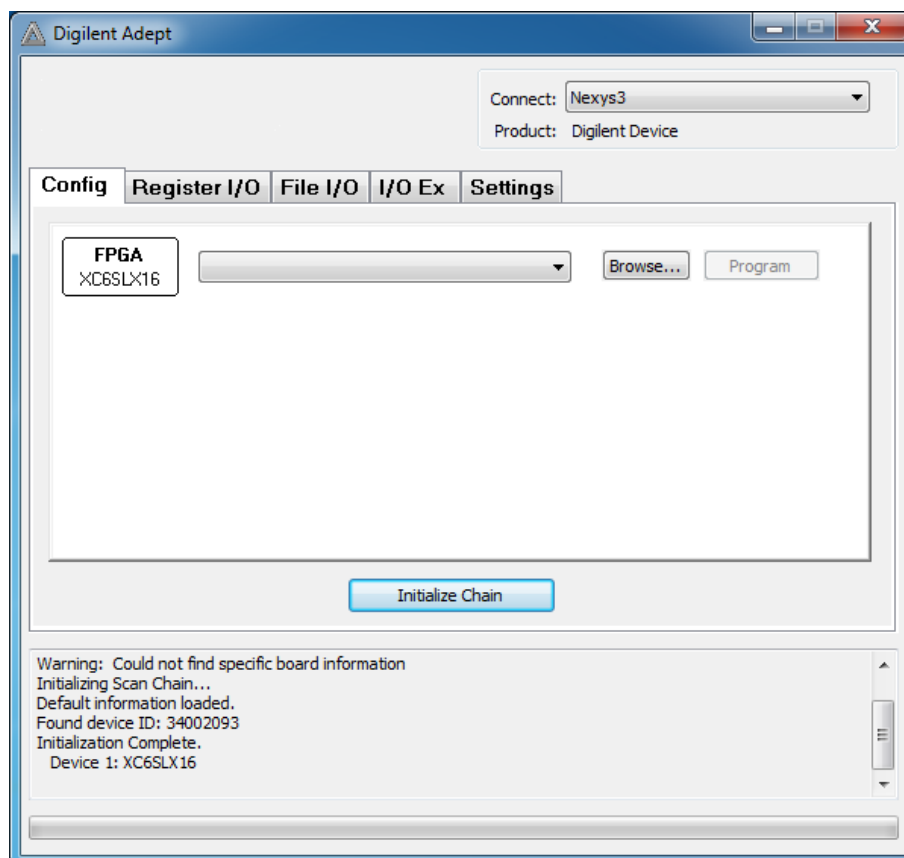


Obrázek 33: Symboly, a) „fajfka“, b) varování, c) chyba

V dalším kroku je potřeba vytvořit konfigurační soubor. Ten může mít různý formát v závislosti na konfiguraci. V tomto případě se vytvoří do složky s projektem soubor s příponou .bit. Soubor se vytvoří spuštěním „**Generate Programing File**“ který lze vidět na Obrázku 21.

Následuje konfigurace FPGA. K tomu je použit program Adept od firmy Digilent. Nejprve je nutné připojit vývojovou desku k počítači přes USB konektor (USB PROG) nacházející se v levém horním rohu desky. Nad tímto konektorem se nachází spínač pro zapnutí desky (červená LED dioda indikuje zapnutí). Dalším krokem je spuštění programu Adept. V pravém horním rohu se nachází záložka „**Connect**“, kde je vypsána deska Nexys 3. Program vypíše v dialogovém oknu hlášení o inicializaci desky.





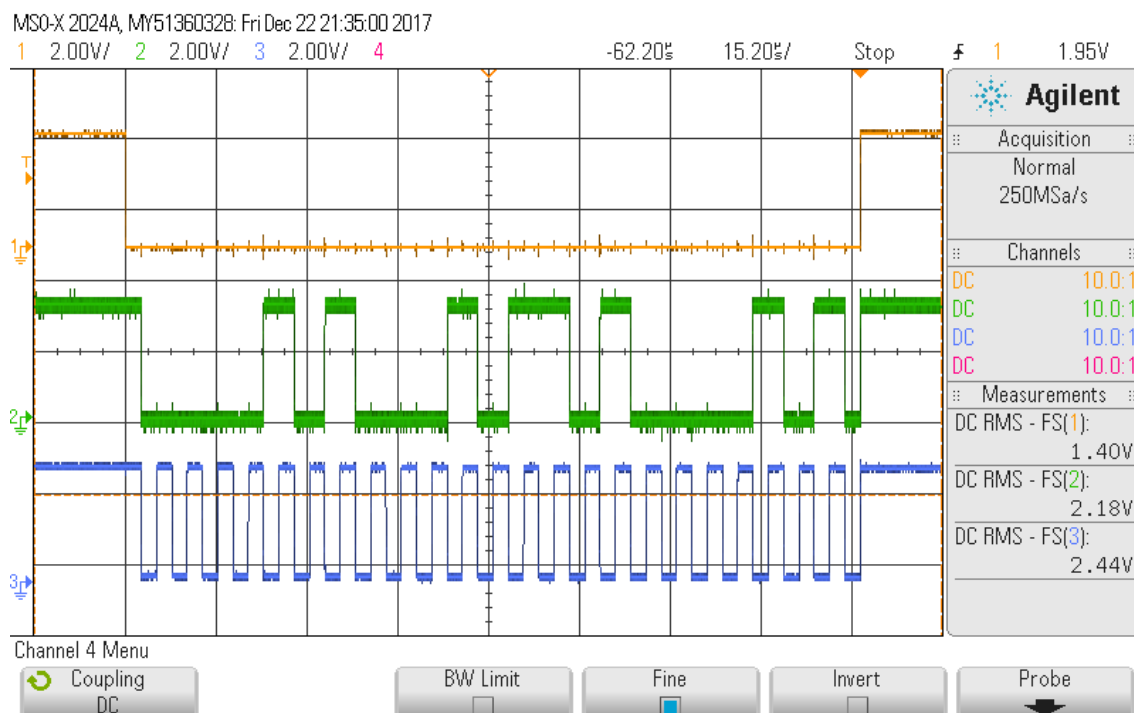
Obrázek 34: Program Digilent Adept

Poté se stiskne tlačítko „**Browse**“ a vyhledá se požadovaný soubor s příponou .bit. Poslední krok je stisknutí tlačítka „**Program**“, tím se nahraje do FPGA konfigurační soubor a v dialogovém okně se vypíše hlášení, zda je soubor úspěšně nahrán. V případě, že nahrání proběhlo v pořádku se vypíše „**Programming Successful**“.

### 9.1 Ověření funkčnosti a zhodnocení výsledků

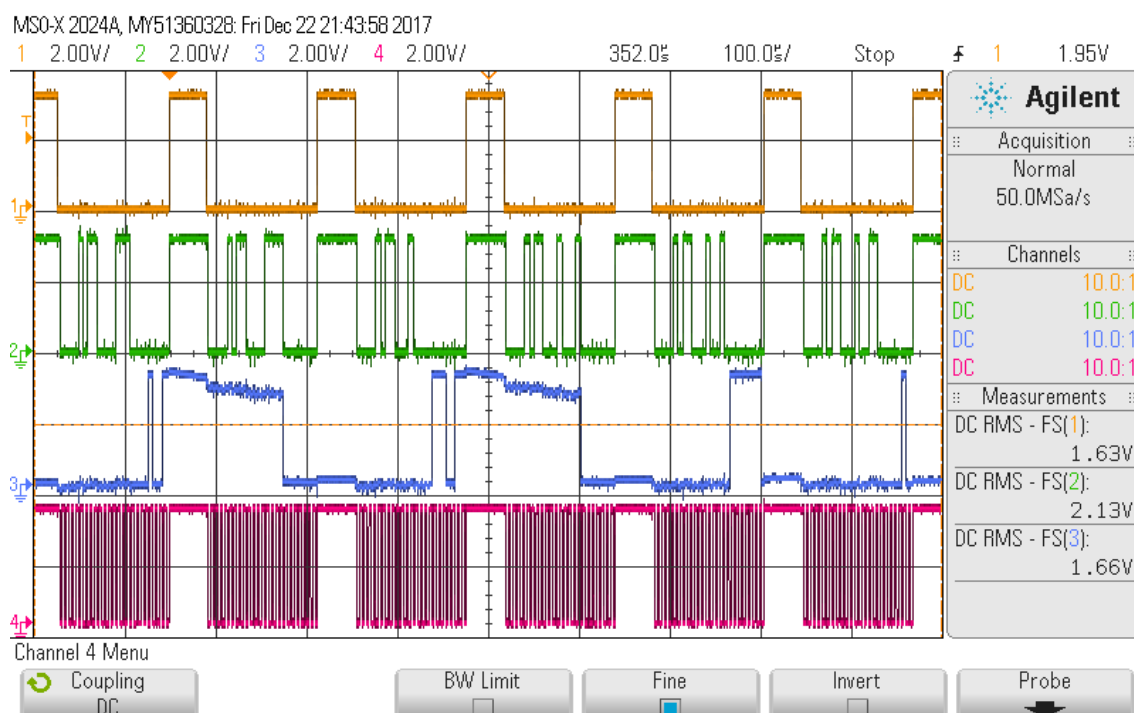
Ověření funkčnosti bylo dvojí, nejprve se ověřovala funkčnost vizuálně (video v příloze), a poté se ověřovala komunikace s akcelerometrem pomocí osciloskopu. Pro měření byly použity 4 kanály osciloskopu, které byly podle příslušného měření zapojeny k sériové sběrnici SPI. Sondy byly přivedeny k vodičům CS, MOSI, MISO, SCLK. Měření bylo provedeno pomocí speciální funkce osciloskopu „Serial“.

Jako první byla měřena konfigurace akcelerometru, která proběhne při zapnutí obvodu. Konfigurace se provádí zápisem hodnot do registru akcelerometru. Na následujícím obrázku je zobrazen signál CS (žlutě), který změnou své hodnoty na log. 0 indikuje začátek přenosu dat. Dále je zde vidět signál MOSI (zeleně) a hodinový signál SCLK (modře).<sup>[10]</sup>

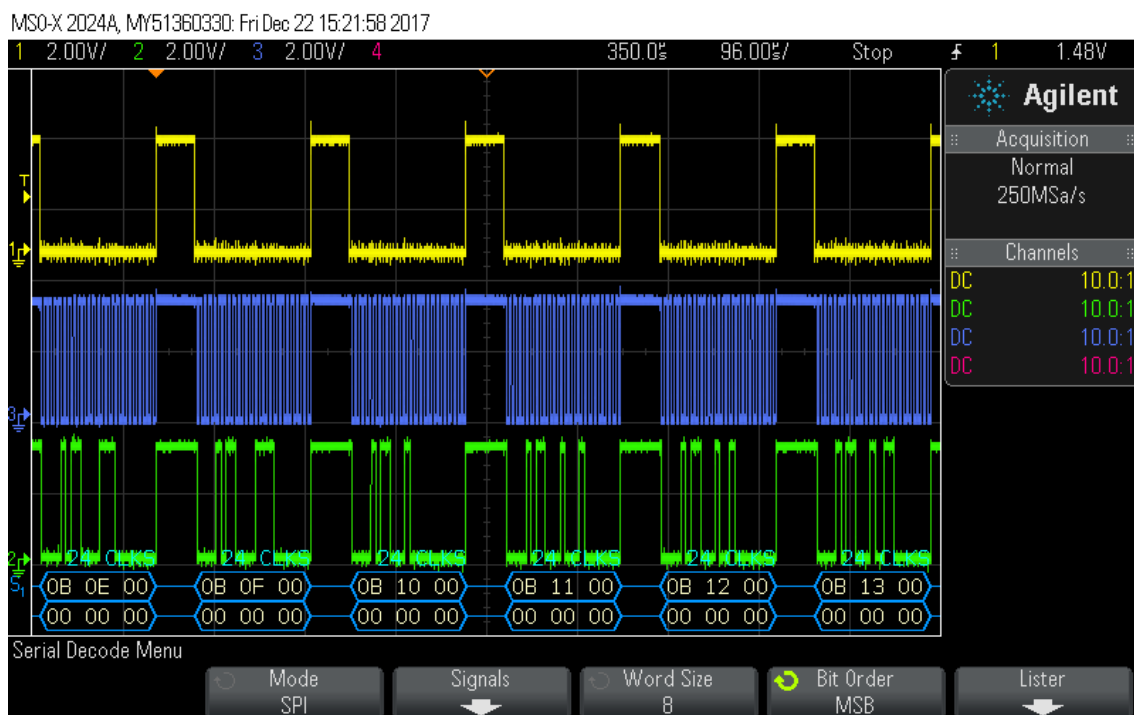


Obrázek 35: Detailní zobrazení prvního zápisu do registru akcelerometru

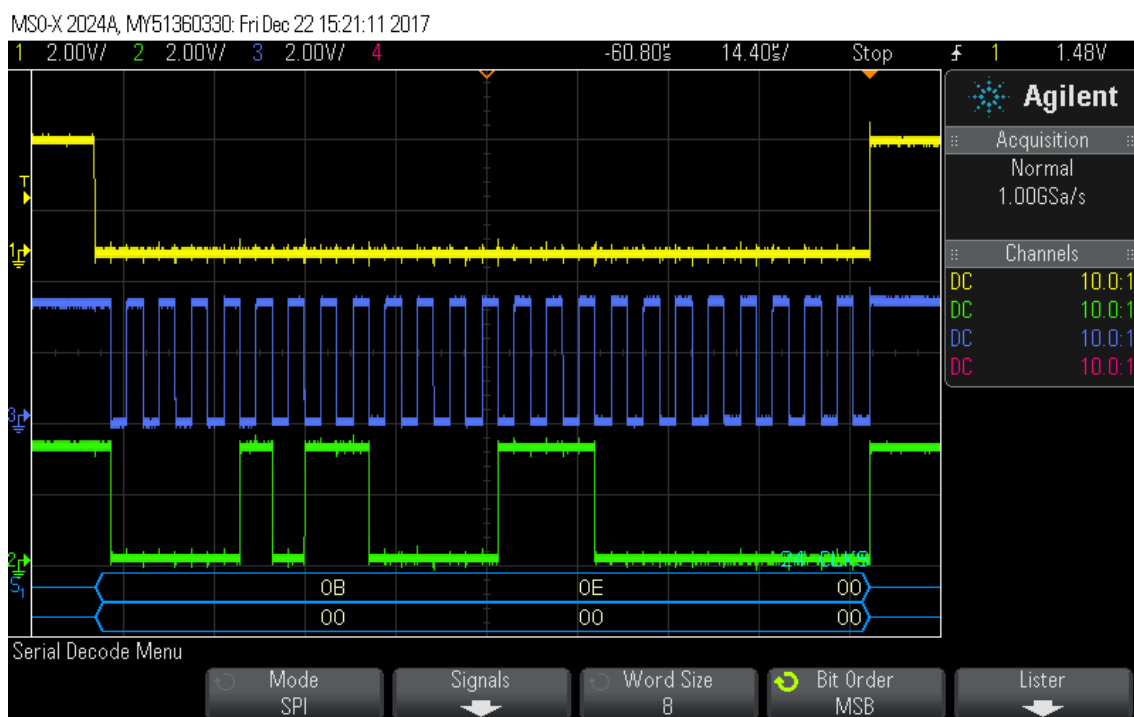
Dalším měřením bylo čtení dat z akcelerometru, které se opakovaně provádí po nakonfigurování. Signál CS je žlutý, růžový je SCLK, zelený signál (MOSI) ukazuje data posílaná do akcelerometru a modře jsou čtená data z akcelerometru (vodič MISO).



Obrázek 36: Průběh čtení dat z akcelerometru



Obrázek 37: Příkaz pro čtení z akcelerometru



Obrázek 38: Detailní zobrazení příkazu pro čtení z akcelerometru

Na tomto obrázku můžeme vidět detail čtení z prvního registru. Zahájení přenosu je signálem CS do log. 0, což spustí komunikaci. Hodnota 0B znamená čtení z registru a další hodnota 0E říká, z jaké adresy se bude číst. Přenos je ukončen navrácením signálu CS do log. 1.

## Závěr

Cílem práce bylo vytvoření výukového modelu s využitím programovatelné logiky FPGA a modulu PmodACL2 a PmodSTEP, který ovládá krokový motor.

Teoretická část obsahuje obecné seznámení s technologií a komponentami využitými v praktické části bakalářské práce.

V demonstrační úloze je modul Pmod ACL2 umístěn na předloktí uživatele a v případě pohybu ruky se v reálném čase pohne i model ruky upevněný ke hřídeli krokového motoru. Výukový model byl úspěšně vytvořen a jeho funkce byla ověřena. Ověření bylo provedeno měřením dat z akcelerometru pomocí osciloskopu a vizuálním posudkem. Akcelerometr je tříosý a je nastaven na měřicí rozsah  $\pm 2g$  při citlivosti 1 mg/LSB. Měřená data jsou ve tvaru dvojkového doplňku o šířce 12 bitů. Z těchto dat se vypočítává klouzavý průměr, aby se odstranil šum vzniklý při měření. Dále se z hodnot dvou os vypočítává úhel natočení akcelerometru, který dává informaci o aktuální poloze paže a z této informace se vyhodnocuje, zda se krokový motor pootočit. Model se může dále rozvíjet a sloužit jako výukový materiál. V příloze je vytvořena laboratorní úloha, ve které budou studenti blíže seznámení s modulem PmodSTEP, se kterým budou pracovat. Práce je velmi obsáhlá a může sloužit jako podklad pro další laboratorní úlohy.

## Seznam použité literatury

- [1] ŘEZÁČ, Kamil. Krokové motory : princip funkce, metody řízení. In: *Robotika.cz* [online]. 2002-10-28. [cit. 2017-06-30]. Dostupné z: <https://robotika.cz/articles/steppers/cs>.
- [2] Princip - Krokový motor. *Pohonnatechnika.cz* [online]. 2018 [cit. 2018-04-29]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/krokovy-motor>
- [3] SLINTÁK, Vlastimil. Úvod do PLD a jazyka VHDL. *UART.CZ Electronics, programming and stuff...* [online]. 2012 [cit. 2018-01-02].
- [4] PARNELL, Karen and Nick MEHTA. *Programmable Logic Design Quick Start Handbook*. 4th ed. [s.l.]: Xilinx Inc., 2003. 225 s.
- [5] Types of PLDs. In: *SlideShare* [online]. LinkedIn Corporation, 2013 [cit. 2018-01-03]. Dostupné z: <https://image.slidesharecdn.com/programmablelogicdevice-130606150852-phapp01/95/programmable-logic-device-pld-2-638.jpg?cb=1370532033>
- [6] SIMPLE PROGRAMMABLE LOGIC DEVICES. *Safari* [online]. Safari Books, 2018 [cit. 2018-04-20]. Dostupné z: <https://www.safaribooksonline.com/library/view/introduction-to-digital/9780470900550/chap1-sec007.html>
- [7] Programmable Logic Devices (PLDs) *Faculty Personal Homepage-KFUPM* [online]. [cit. 2018-04-20]. Dostupné z: <http://faculty.kfupm.edu.sa/COE/about/Lesson6.1.pdf>
- [8] EEPLD Definition. *PC Magazine Encyclopedia* [online]. The Computer Language Company, 2018 [cit. 2018-04-20]. Dostupné z: <https://www.pcmag.com/encyclopedia/term/42402/eepld>
- [9] CPLD Architecture. In: *SlideShare* [online]. LinkedIn Corporation, 2013 [cit. 2018-01-03]. Dostupné z: <https://image.slidesharecdn.com/lec22-130220025308-phpapp02/95/cplds-22-638.jpg?cb=1362136123>
- [10] What is an FPGA? *XILINX* [online]. XILINX, 2018 [cit. 2018-01-02]. Dostupné z: <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>
- [11] WHITE, Robert. FPGA Architecture. In: *RF Neulink by Raveon Technologies* [online]. [cit. 2018-01-03]. Dostupné z: <http://www.rfneulink.com/image/WP-FPGA-diagram.jpg>
- [12] DEKKER, Rob. What's the Difference Between VHDL, Verilog, and SystemVerilog?. *Electronic Design* [online]. Informa USA, 2018, 2014 [cit. 2018-04-20]. Dostupné z: <http://www.electronicdesign.com/what-s-difference-between/what-s-difference-between-vhdl-verilog-and-systemverilog>
- [13] VHDL. *IT SLOVNÍK.CZ* [online]. IT-Slovník.cz team, 2017 [cit. 2018-01-02]. Dostupné z: <https://it-slovník.cz/pojem/vhdl>

- [14] Nexys 3 Spartan-6 FPGA Trainer Board. *DIGILENT A National Instruments Company* [online]. DIGILENT [cit. 2018-04-20]. Dostupné z: <https://store.digilentinc.com/nexys-3-spartan-6-fpga-trainer-board-limited-time-see-nexys4-ddr/>
- [15] Pmod Expansion ports. In: *Digilent A National Instruments Company* [online]. DIGILENT [cit. 2018-04-29]. Dostupné z: <https://reference.digilentinc.com/reference/programmable-logic/netfpga-1g-cml/reference-manual>
- [16] Digilent Pmod™ Interface Specification. *DIGILENT A National Instruments Company* [online]. Washington: DIGILENT, 2011 [cit. 2018-01-03].
- [17] Pmod I2S: Stereo Audio Output. In: *DIGILENT A National Instruments Company* [online]. DIGILENT [cit. 2018-04-20]. Dostupné z: <https://store.digilentinc.com/pmod-i2s-stereo-audio-output/>
- [18] Pmod AD2: 4-channel 12-bit A/D Converter. In: *DIGILENT A National Instruments Company* [online]. DIGILENT [cit. 2018-04-20]. Dostupné z: <https://store.digilentinc.com/pmod-ad2-4-channel-12-bit-a-d-converter/>
- [19] Pmod ACL2: 3-axis MEMS Accelerometer. *Digilent A National Instruments Company* [online]. DIGILENT [cit. 2018-04-14]. Dostupné z: <https://store.digilentinc.com/pmod-acl2-3-axis-mems-accelerometer/>
- [20] Pmod ACL2 Reference Manual. *Digilent A National Instruments Company* [online]. DIGILENT [cit. 2018-04-14]. Dostupné z: <https://reference.digilentinc.com/reference/pmod/pmodacl2/reference-manual>
- [21] Analog devices, Inc. *Data Sheet ADXL362* [on-line]. [cit. 2018-04-14]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL362.pdf>
- [22] Pmod STEP: Stepper Motor Driver. *DIGILENT A National Instruments Company* [online]. DIGILENT [cit. 2018-04-29]. Dostupné z: <https://store.digilentinc.com/pmod-step-stepper-motor-driver>
- [23] KALAŠ, Michal. Notebook. In: *Metodický portál inspirace a zkušenosti učitelů* [online]. 2011 [cit. 2018-04-06]. Dostupné z: <http://wiki.rvp.cz/@api/deki/files/19345/=notebook.jpg>
- [24] Stepper Clipart. In: *Clipart Library* [online]. Clipart Library, 2016 [cit. 2018-04-06]. Dostupné z: <http://clipart-library.com/clipart/328684.htm>
- [25] Arm Clipart. In: *Clipart Library* [online]. Clipart Library, 2016 [cit. 2018-04-06]. Dostupné z: <http://clipart-library.com/clipart/224843.htm>
- [26] Minebea PM42L-48. In: *MinebeaMitsumi* [online]. Langen (Hessen): Minebea Co. [cit. 2018-01-03]. Dostupné z: <http://www.nmbtc.com/content/pdfs/PM42L-048%20Standard.pdf>

- [27] MINEBEA NEMA STEPPER MOTOR 17PM-K502. In: *CNCStore* [online]. [cit. 2018-04-29]. Dostupné z: <http://cncsuperstore.com/image/cache/data/Products/motors/17PM-K502-P1ST/A4-228x228.JPG>
- [28] 4-Wire Stepper Motor. In: *Electronic Stepper On Store* [online]. 2012 [cit. 2018-04-29]. Dostupné z: <http://ecx.images-amazon.com/images/I/41f25jYh2kL.jpg>
- [29] Schrittmotor Stepper Motor Minebea 34PM - C101. In: *PicClick DE* [online]. [cit. 2018-04-29]. Dostupné z: [https://www.picclickimg.com/d/l400/pict/173053000776\\_/Schrittmotor-Stepper-Motor-Minebea-34PM-C101-18.jpg](https://www.picclickimg.com/d/l400/pict/173053000776_/Schrittmotor-Stepper-Motor-Minebea-34PM-C101-18.jpg)
- [30] Precision Step Motors, Inc. *MNB corporation* [on-line]. [cit. 2018-04-29]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL362.pdf>
- [31] Chip. In: *Adobe Stock* [online]. [cit. 2018-04-06]. Dostupné z: [https://t3.ftcdn.net/jpg/00/99/09/80/500\\_F\\_99098095\\_sZRMl1P1eBBFCpyRciLwb02xuJ2rRLdA.jpg](https://t3.ftcdn.net/jpg/00/99/09/80/500_F_99098095_sZRMl1P1eBBFCpyRciLwb02xuJ2rRLdA.jpg)
- [32] MACÍČEK, Martin. *Měření s tříosým akcelerometrem prostředky programovatelné logiky* [online]. Ostrava, 2015 [cit. 2018-01-03]. Bakalářská práce. VŠB – Technická univerzita Ostrava Fakulta elektrotechniky a informatiky Katedra kybernetiky a biomedicínského inženýrství. Vedoucí práce Ing. Vladimír Kašík, Ph.D.
- [33] Filter architecture. In: *LET'S ELECTRONIC* [online]. [cit. 2018-04-30]. Dostupné z: <https://4.bp.blogspot.com/-MUuOqkbUSgQ/WE816dWRI1I/AAAAAAAAABTU/hMwkhaXpKZs8N0EnwbOzjICTYDFFGZ7vACLcB/s640/Sans>
- [34] Technické indikátory: Klouzavé průměry. *Daytrade.cz* [online]. [cit. 2018-04-30]. Dostupné z: <http://daytrade.cz/klouzave-prumery/>
- [35] LogiCORE IP CORDIC v4.0. *Cordic\_ds249 Data Sheet* [online]. Inc. XILINX, 2011 [cit. 2018-04-30]. Dostupné z: [https://www.xilinx.com/support/documentation/ip\\_documentation/cordic\\_ds249.pdf](https://www.xilinx.com/support/documentation/ip_documentation/cordic_ds249.pdf)

## Seznam příloh

### Interní přílohy

Příloha A	Laboratorní úloha PmodSTEP (12 stran) .....	i
Příloha B	Laboratorní úloha PmodACL2_PmodSTEP (5 stran) .....	xiv
Příloha C	RTL schéma entity Kyberneticky_model (1 strana) .....	xx
Příloha D	RTL schéma komponenty ACL_SPI (1 strana) .....	xxi

### Externí přílohy – elektronické

Příloha E	CD/DVD .....	xxii
-----------	--------------	------



## **A Laboratorní úloha PmodSTEP**

### **A.1 Cíl**

V této laboratorní úloze:

- se seznámíte s modulem PmodSTEP,
- se seznámíte s krokovým motorem,
- se naučíte ovládat bipolární krokový motor.

### **A.2 Zadání**

1. Prostudujte si VHDL kód projektu Kyberneticky\_model\_PmodSTEP.
2. Zjistěte jak se chová motor při různé hodnotě odebíraného proudu.
3. Zrychlete otáčení krokového motoru, aby byla rychlost dvakrát větší než původní a přidejte spínač na změnu směru otáčení hřídele motoru.

### **A.3 Předpokládané znalosti**

Pro tuto úlohu se vyžaduje nastudování:

- datasheet modul PmodSTEP,
- datasheet Krokový motor 17PM-K502-G2ST MINEBEA.

### **A.4 Použité vybavení**

- vývojová deska NEXYS 3,
- modul PmodSTEP,
- krokový motor 17PM-K502-G2ST MINEBEA.

### **A.5 Teoretický rozbor**

#### **A.5.1 Konstrukce krokových motorů**

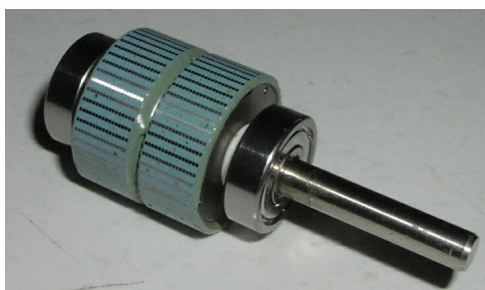
Krokový motor se skládá ze statického statoru a pohyblivého rotoru. Rotor se dále skládá z hřídele usazené na kuličkových ložiscích a prstence permanentních magnetů. Stator se skládá ze sady cívek, kde pólové nástavce jsou vroubkovány se stejnou roztečí, jako je rozteč magnetů na rotoru. Díky tomu získáme vyšší přesnost motoru při stejném počtu cívek.[1]



Obrázek 1: Celkový pohled na krokový motor[1]



Obrázek 2: Stator[1]



Obrázek 3: Rotor[1]

### Princip krokového motoru

Základním principem krokového motoru je procházení proudu cívkou statoru, díky čemu se vytvoří magnetické pole, které přitáhne opačný pól magnetu rotoru. Zapojováním různých cívek dosáhneme rotačního pohybu hřídele rotoru.

Přechodové magnetické jevy ovlivňují maximální rychlost otáčení motoru (stovky kroků za sekundu, záleží podle konkrétního motoru). Při překročení této rychlosti nebo při velké zátěži dojde ke ztrátě kroků. Existuje několik variant řízení krokového motoru, avšak nejprve

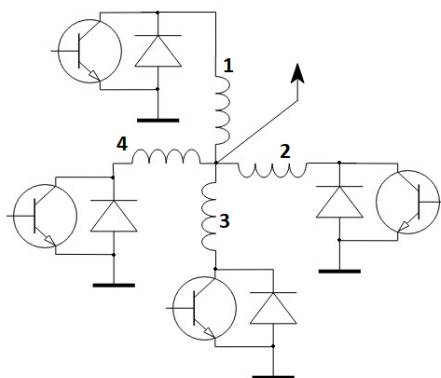
je potřeba si určit požadovaný kroutící moment, přesnost nastavení polohy (kolik stupňů je jeden krok) a odběr proudu. Podle materiálu rotoru se krokové motory rozdělují na několik typů, buď může být tvořen železným jádrem tzv VR (variabilní reluktance), permanentními magnety (PM) a nebo může být hybridní, což je kombinace VR a PM.[1], [2]

### Způsob řízení krokového motoru

Existuje několik variant řízení motoru, různé způsoby jsou vypsány v následujících kapitolách.

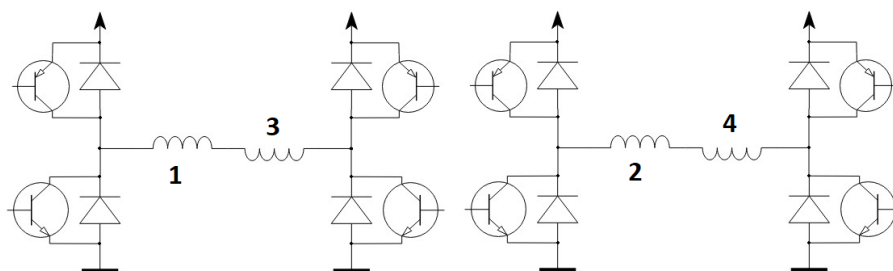
#### a) Unipolární a bipolární řízení

Unipolární řízení je specifické tím, že proud prochází v jednom okamžiku pouze jednou cívkou a motor s tímto buzením má nejmenší odběr, ale také má nejmenší kroutící moment. Výhodou tohoto řízení je jednoduché zapojení (stačí jen jeden tranzistor na každou cívku).



Obrázek 4: Schéma unipolárního řízení krokového motoru[1]

Naproti tomu bipolární řízení se vyznačuje tím, že proud prochází vždy dvěma protilehlými cívkami. Cívky jsou zapojeny tak, že mají opačnou polaritu. Tímto zapojením dosáhneme většího kroutícího momentu. Nevýhodou je vyšší spotřeba (krokový motor se i více zahřívá). Náročnější je také i zapojení, jsou zapotřebí 2 H-můstky (pro každou větev jeden).[1]



Obrázek 5: Schéma bipolárního řízení krokového motoru[1]

### b) Jednofázové versus dvoufázové řízení

Jednofázové řízení spočívá v tom, že magnetické pole generuje pouze jedna cívka (při unipolárním buzení) případně dvojice cívek (při bipolárním buzení).

Dvoufázové řízení funguje tak, že dvě sousední cívky mají stejně orientované magnetické pole. Získáme tím vyšší kroutící moment a dvojnásobnou spotřebu oproti jednofázovému řízení.[1]

### c) Řízení s plným a polovičním krokem

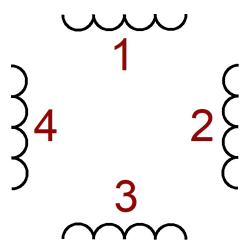
Řízení s plným krokem je takový typ řízení, kdy se provede na jednu otáčku tolik kroků, kolik zubů má stator daného motoru. Tohoto krokování dosáhneme použitím jak unipolární tak bipolární metodou buzení.

U řízení s polovičním krokem se jedná o střídání kroků s jednofázovým a dvoufázovým řízením, čímž dosáhneme dvojnásobné přesnosti.[1]

### **Řízení krokového motoru**

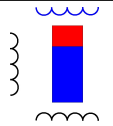
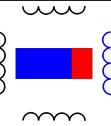
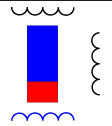
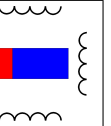
Pro zjednodušení budeme uvažovat o krokovém motoru se čtyřmi kroky na otáčku.

Černou barvou je označena cívka, která je bez proudu (v tabulce je zaznačena jako „0“). Modrá cívka přitahuje červený konec magnetu rotoru (v tabulce je označena jako „-“). Magnetické pole červené cívky přitahuje modrý konec magnetu (v tabulce označené jako „+“).[1]

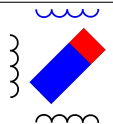
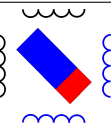
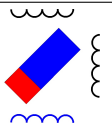
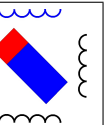


Obrázek 6: Rozložení cívek krokového motoru[1]

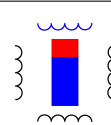
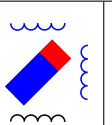
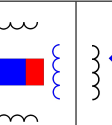
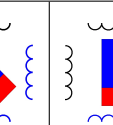
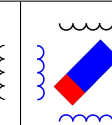
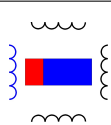
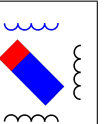

Tabulka 1: Unipolární jednofázové řízení s plným krokem[1]

				
Cívka 1	-	0	0	0
Cívka 2	0	-	0	0
Cívka 3	0	0	-	0
Cívka 4	0	0	0	-

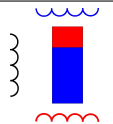
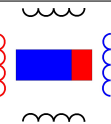
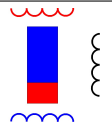
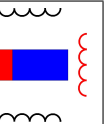
Tabulka 2: Unipolární dvoufázové řízení s plným krokem[1]

				
Cívka 1	-	0	0	-
Cívka 2	-	-	0	0
Cívka 3	0	-	-	0
Cívka 4	0	0	-	-

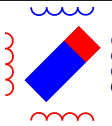
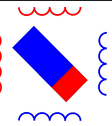
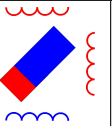
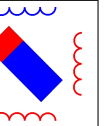
Tabulka 3: Unipolární řízení s polovičním krokem[1]

								
Cívka 1	-	-	0	0	0	0	0	-
Cívka 2	0	-	-	-	0	0	0	0
Cívka 3	0	0	0	-	-	-	0	0
Cívka 4	0	0	0	0	0	-	-	-

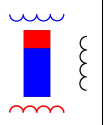
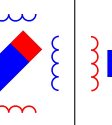
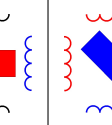
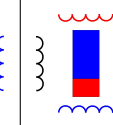
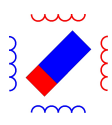
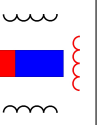
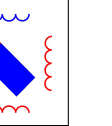

Tabulka 4: Bipolární jednofázové řízení s plným krokem[1]

				
Cívka 1	-	0	+	0
Cívka 2	0	-	0	+
Cívka 3	+	0	-	0
Cívka 4	0	+	0	-

Tabulka 5: Bipolární dvoufázové řízení s plným krokem[1]

				
Cívka 1	-	+	+	-
Cívka 2	-	-	+	+
Cívka 3	+	-	-	+
Cívka 4	+	+	-	-

Tabulka 6: Bipolární řízení s polovičním krokem[1]

								
Cívka 1	-	-	0	+	+	+	0	-
Cívka 2	0	-	-	-	0	+	+	+
Cívka 3	+	+	0	-	-	-	0	+
Cívka 4	0	+	+	+	0	-	-	-

### A.5.2 Modul PmodSTEP pro řízení krokových motorů

PmodSTEP komunikuje s HŘD prostřednictvím protokolu GPIO. Tento modul může řídit 4-pinový a současně 6-pinový krokový motor. Krokové motory pracují střídavě napájením cívek na různé polaridy, které indukují krokový motor k otáčení. Detailnější vysvětlení viz kapitola A.5.1.

4-pinové krokové motory pracují pouze v bipolárním uspořádání a vyžadují, aby dva vstupy na každé elektromagnetické cíve byly přivedeny na správné logické napěťové úrovně tak, aby indukovaly proudový tok ve správném směru.

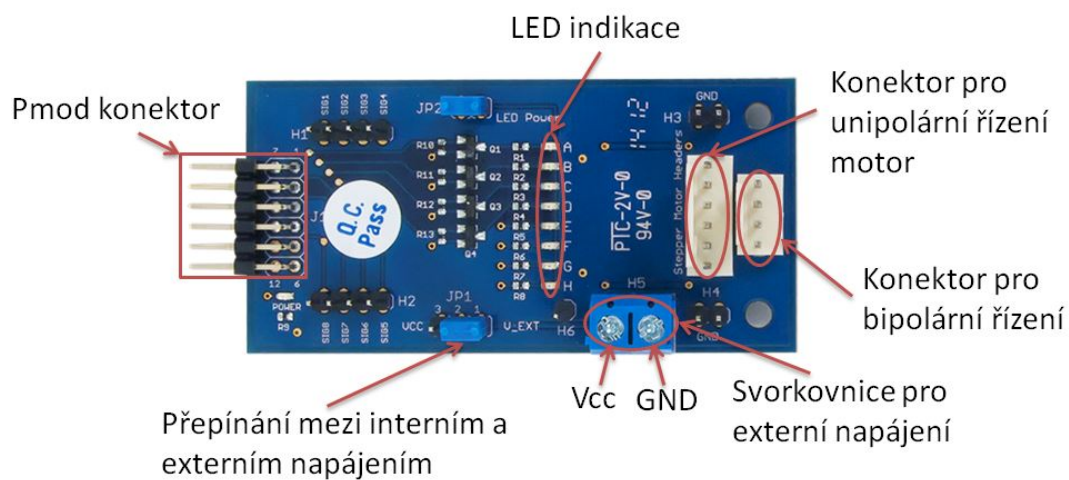
6-pinové krokové motory mohou být orientovány buď pro bipolární nebo unipolární konfiguraci.

PmodSTEP využívá čtyřkanálový ovladač ST L293DD k řízení krokových motorů při vyšších proudech, než systémová deska obvykle poskytuje z jejich logických výstupů. Pro snadné testování a sledování šíření signálů jsou k dispozici externí testovací body a LED diody.[3]

#### Funkce modulu:

- může pohánět 2 motory současně,
- LED indikace buzení výstupů,

- propojka pro externí napájení,
- malá velikost DPS 2,8" × 1,3" (7,1 cm × 3,3 cm),
- 2x 6-pinový konektor Pmod s rozhraním GPIO.[3]



Obrázek 7: Modul PmodSTEP[3]

Tabulka 7: Rozdělení pinů modulu PmodSTEP[3]

Pin	Signál	Popis
1	SIG1	Signál 1
2	SIG2	Signál 2
3	SIG3	Signál 3
4	SIG4	Signál 4
5	GND	Zemní svorka
6	VCC	Napájecí napětí
7	SIG5	Signál 5/Výstup 1 pro krokový motor
8	SIG6	Signál 6/Výstup 2 pro krokový motor
9	SIG7	Signál 7/Výstup 3 pro krokový motor
10	SIG8	Signál 8/Výstup 4 pro krokový motor
11	GND	Zemní svorka
12	VCC	Napájecí napětí

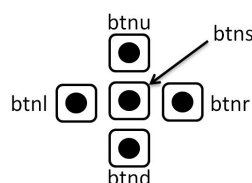
Jakékoliv externí napájení musí být mezi 4,5–36 V, doporučená úroveň napětí je 5 V.[3]

### A.5.3 Projekt Kyberneticky\_model\_PmodSTEP

V následující tabulce je přehled vstupně/výstupních signálů.

Tabulka 8: Přehled I/O signálu projektu Kyberneticky\_model\_PmodSTEP

Název	I/O pin	Směr	Popis
CLK	V10	IN	hodinový signál (100 MHz)
btnd	C9	IN	tlačítko RESET
SW 0	T5	IN	spínač START
JD 4	D12	OUT	výstup na Pmod konektor JA
JD 5	C12	OUT	výstup na Pmod konektor JA
JD 6	F12	OUT	výstup na Pmod konektor JA
JD 7	E12	OUT	výstup na Pmod konektor JA



Obrázek 8: Rozložení tlačítek na vývojové desce Nexys 3

### Popis funkce

Spínačem START se spustí otáčení hřídele krokového motoru po směru hodinových ručiček, vypnutím tohoto spínače dosáhneme pozastavení pohybu. Při stisknutí tlačítka RST se vynuluje dělička a čítač.

### A.5.4 Dělička hodinového signálu

Dělička je napsaná jako proces do kterého vstupují signály CLK a RST (reset). Když dojde ke změně stavu některého z těchto signálů, proces se provede. Dále v procesu můžeme vidět signál count a clk\_krok. Signál count čítá náběžné hrany CLK a clk\_krok je výstupní generovaný signál. Celý proces je kontrolován několika podmínkami. První podmínka nám hlídá, zda nebyl vyžádán reset (zmáčknutí tlačítka). Když dojde ke změně z log. 0 na log. 1, pak dojde nastavení všech signálů do výchozí hodnoty 0. Následující podmínka kontroluje náběžnou hranu CLK. Pokud přijde náběžná hrana a hodnota count nedosáhla hodnoty 4 500 000, tak se k signálu count přičte jednička. Tohle se opakuje, dokud nedosáhne hodnota count maxima, poté se na clk\_krok objeví log. 1 a čítač se vynuluje. Když se celý proces zopakuje, tak se na clk\_krok objeví log 0. Tím dojde ke generování signálu se skoro libovolným kmitočtem .



```

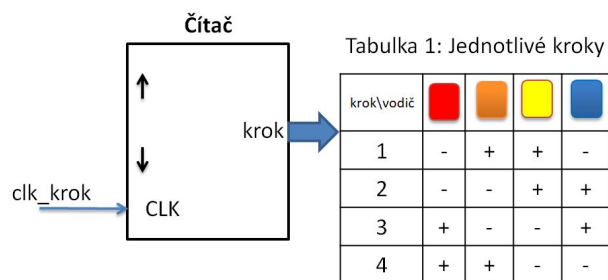
delicka: process (CLK, RST)
begin
if RST = '1' then
count <= (others => '0');
clk_krok <= '0';
elsif Rising_Edge (CLK) then
if count = 4000000 then
count <= (others => '0');
clk_krok <= not clk_krok;
else
count <= count + 1;
end if;
end if;
end process delicka;

```

Obrázek 9: Část VHDL kódu pro děličku hodinového signálu

### A.5.5 Čítač a vyhodnocení kroků

Z děličky vstupuje do čítače signál clk\_krok a vystupuje signál krok, který má dvoubitovou šířku tudíž čítá od 0 do 3 (v tabulce od 1 do 4). Dle aktuální hodnoty kroku se provede odpovídající fáze krokování viz tabulka 1.



Obrázek 10: Část VHDL kódu pro děličku hodinového signálu

Další částí je 2/4 dekodér, pro přiřazení jednotlivé fáze kroku odpovídajícímu pinu konektoru Pmod.

```

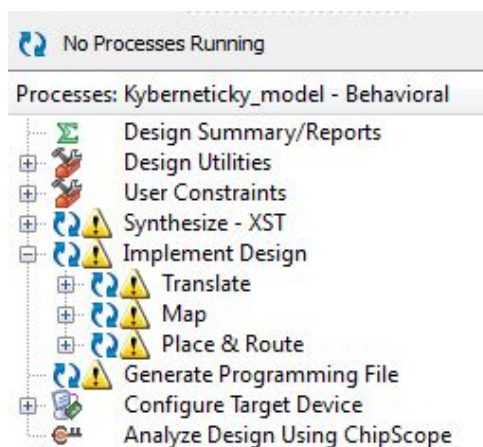
with krok select JD <=
"1100" when "00",
"1001" when "01",
"0011" when "10",
"0110" when "11";

```

Obrázek 11: VHDL kód pro vyhodnocení kroků

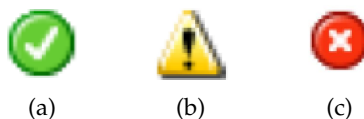
### A.5.6 Implementace a oživení

Implementaci zahájíme spuštěním „**Implement Design**“.



Obrázek 12: *Implementace*

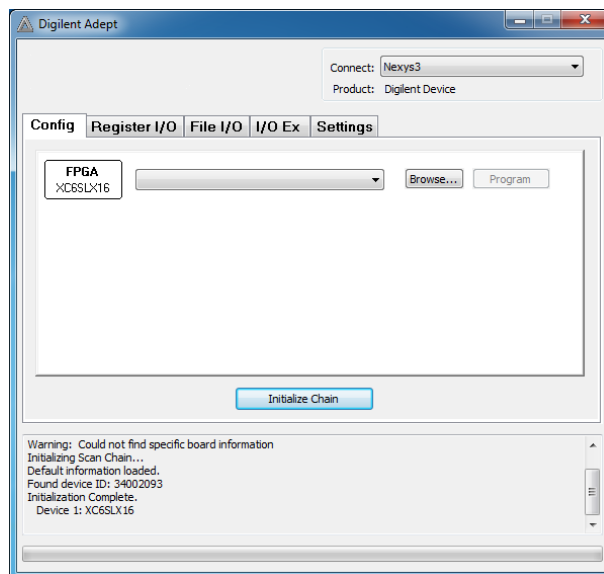
Po ukončení implementace se u každé části můžou zobrazit tři různé symboly, když byl nějaký krok úspěšný, tak se zobrazí symbol „fajfky“ (OK = v pořádku), . V případě nevhodného návrhu se zobrazí symbol vykřičník (warning = varování), toto varování neukončí implementaci, ale je vhodné si přečíst hlášení. Pokud se zobrazí symbol křížku (error = chyba), tak se ukončí aktuální krok implementace a je nutné chybu odstranit.



Obrázek 13: *Symbols, a) „fajfka“, b) varování, c) chyba*

V dalším kroku je potřeba vytvořit konfigurační soubor. Ten může mít různý formát v závislosti na konfiguraci. V tomto případě se vytvoří do složky s projektem soubor s příponou .bit. Soubor se vytvoří spuštěním „**Generate Programing File**“ který lze vidět na Obrázku 21.

Následuje konfigurace FPGA. K tomu je použit program Adept od firmy Digilent. Nejprve je nutné připojit vývojovou desku k počítači přes USB konektor (USB PROG) nacházející se v levém horním rohu desky. Nad tímto konektorem se nachází spínač pro zapnutí desky (červená LED dioda indikuje zapnutí). Dalším krokem je spuštění programu Adept. V pravém horním rohu se nachází záložka „**Connect**“, kde je vypsána deska Nexys 3. Program vypíše v dialogovém okně hlášení o inicializaci desky.



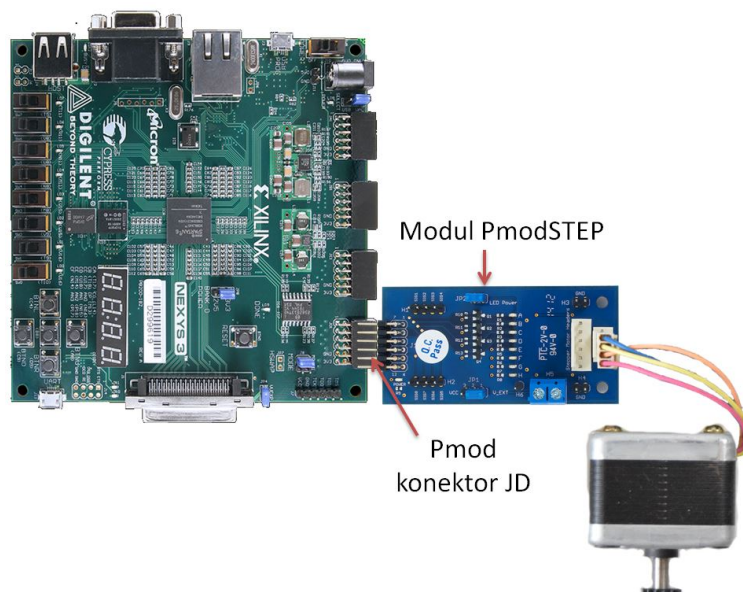
Obrázek 14: Program Digilint Adept

Poté se stiskne tlačítko „**Browse**“ a vyhledá se požadovaný soubor s příponou .bit. Poslední krok je stisknutí tlačítka „**Program**“, tím se nahraje do FPGA konfigurační soubor a v dialogovém okně se vypíše hlášení, zda je soubor úspěšně nahrán. V případě, že nahrání proběhlo v pořádku se vypíše „**Programming Successful**“.

## A.6 Pracovní postup

### A.6.1 Postup k bodu č.1

1. Zapojte modul PmodSTEP do desky NEXYS 3 podle následujícího obrázku.



Obrázek 15: Celkový pohled na model

2. Spusťte program ISE Design Suite.



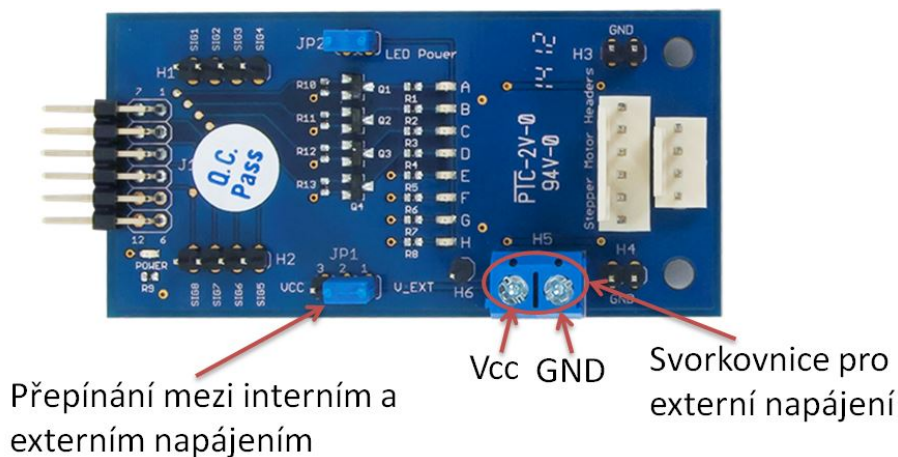
Obrázek 16: Ikona programu ISE Design Suite

3. Otevřete projekt Kyberneticky\_model.

File => Open Project => Otevřete složku projektu Kyberneticky\_model => Kyberneticky\_model.xise

### A.6.2 Postup k bodu č.2

1. Připojte do modulu PmodSTEP externí napájení dle obrázku 17.



Obrázek 17: Znázornění napájení na modulu PmodSTEP

2. Zvolte úroveň napětí 5 V a omezte proud hodnotu na minimum.
3. Spusťte pohyb a pomalu zvyšujte hodnotu proudu do **maximální hodnoty 0,6 A !!!**
  - Sledujte a zhodnoťte jaký vliv má omezení proudu na krouticí moment krokového motoru.
  - K čemu dojde, když pevně uchopíme hřídel motoru a budeme se silou snažit hřídel udržet na místě.

### A.6.3 Postup k bodu č.3

1. Změňte v děliči hodnotu count tak, aby byla rychlost otáčení hřídele motoru dvojnásobná.
2. Připojte další spínač, který bude určovat směr otáčení hřídele.

## Seznam použité literatury

- [1] ŘEZÁČ, Kamil. Krokové motory : princip funkce, metody řízení.In: *Robotika.cz* [online]. 2002-10-28. [cit. 2017-06-30]. Dostupné z: <https://robotika.cz/articles/steppers/cs>.
- [2] Princip - Krokový motor. *Pohonnatechnika.cz* [online]. 2018 [cit. 2018-04-29]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/krokovy-motor>
- [3] Pmod STEP: Stepper Motor Driver. *DIGILENT A National Instruments Company* [online]. DIGILENT [cit. 2018-04-29]. Dostupné z: <https://store.digilentinc.com/pmod-step-stepper-motor-driver>

## **B Laboratorní úloha 2**

### **B.1 Cíl**

V této laboratorní úloze:

- se seznámíte s modulem PmodACL2,
- se naučíte ovládat bipolární krokový motor reagující na data z akcelerometru.

### **B.2 Zadání**

1. Prostudujte si VHDL kódy projektu `Kybernetick_model_PmodACL2_PmodSTEP`.
  - Zaměřte se na formát dat vystupujících z modulu PmodACL2.
2. Vytvořte podmínku, kde bude směr otáčení závislý na natočení modulu PmodACL2.

### **B.3 Předpokládané znalosti**

Tato laboratorní úloha navazuje na laboratorní úlohu PmodSTEP, předpokládá se, že student již pracoval na této laboratorní úloze.

**Pro tuto úlohu se vyžaduje nastudování:**

- datasheet modulu PmodSTEP,
- datasheet modulu PmodACL2,
- datasheet Krokový motor 17PM-K502-G2ST MINEBEA,
- BP: Kybernetický výukový model s krokovými motory – laboratorní úloha.
  - kapitola Komponenta Signal\_START,
  - kapitola Komunikace s akcelerometrem,
  - kapitola Komponenta ACL\_SPI.

### **B.4 Použité vybavení**

- vývojová deska NEXYS 3,
- modul PmodACL2,
- modul PmodSTEP,
- krokový motor 17PM-K502-G2ST MINEBEA.

## B.5 Teoretický rozbor

### B.5.1 Snímaná data

Snímaná data je ve tvaru dvojkového doplnku. Jedná se o způsob kódování celých čísel (kladných i záporných). V následující tabulce je příklad čtyřbitového čísla.[1]

Pozornost zaměřte na bit vlevo (MSB = most significant bit), při kladném čísle nebo nule je jeho hodnota 0, ale v případě záporného čísla je 1.

Tabulka 9: Znáznornění tvaru dvojkového doplnku

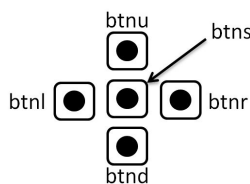
Dvojkový doplněk	Desítková soustava
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
<b>0000</b>	<b>0</b>
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

### B.5.2 Projekt Kybernetický\_model\_PmodACL2\_a\_PmodSTEP

V následující tabulce je přehled vstupně/výstupních signálů entity Kybernetický\_model.

Tabulka 10: Přehled I/O signálu

Název	I/O pin	Směr	Popis
CLK	V10	IN	hodinový signál (100 MHz)
btnd	C9	IN	tlačítko RESET
SW 0	T5	IN	spínač START
LED 0	U16	OUT	LED indikace
JA 0	T12	OUT	SS = slave select
JA 2	V12	OUT	SDO = slave data out
JA 3	N10	IN	SDI = slave data in
JA 4	P11	OUT	SCLK = slave CLK
JD 4	D12	OUT	výstup na Pmod konektor JD
JD 5	C12	OUT	výstup na Pmod konektor JD
JD 6	F12	OUT	výstup na Pmod konektor JD
JD 7	E12	OUT	výstup na Pmod konektor JD



Obrázek 18: Rozložení tlačítek na vývojové desce Nexys 3

#### Popis funkce

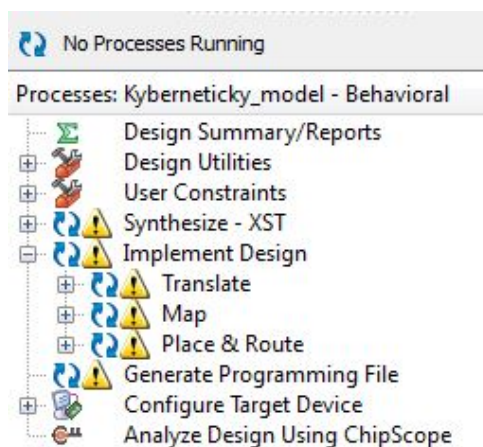
Po nahrání kódu do vývojové desky Nexys 3 začne komunikace s modulem akcelerometru skze SPI. Nejprve se provede konfigurace a poté jsou načítána data, která jsou dále upravována (další laboratorní úloha bude na komunikování s modulem PmodACL2). Do komponenty data\_evaluation přichází už upravená data ve formě dvojkového doplňku.[2]

Spínačem START se spustí otáčení hřídele krokového motoru po směru hodinových ručiček, vypnutím tohoto spínače dosáhneme pozastavení pohybu. Při stisknutí tlačítka RST dojde k resetu celého obvodu, po uvolnění tlačítka se znovu provede konfigurace modulu akcelerometru.



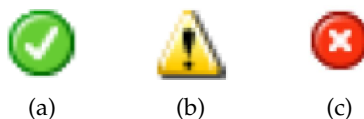
### B.5.3 Implementace a oživení

Implementaci zahájíme spuštěním „**Implement Design**“.



Obrázek 19: Implementace

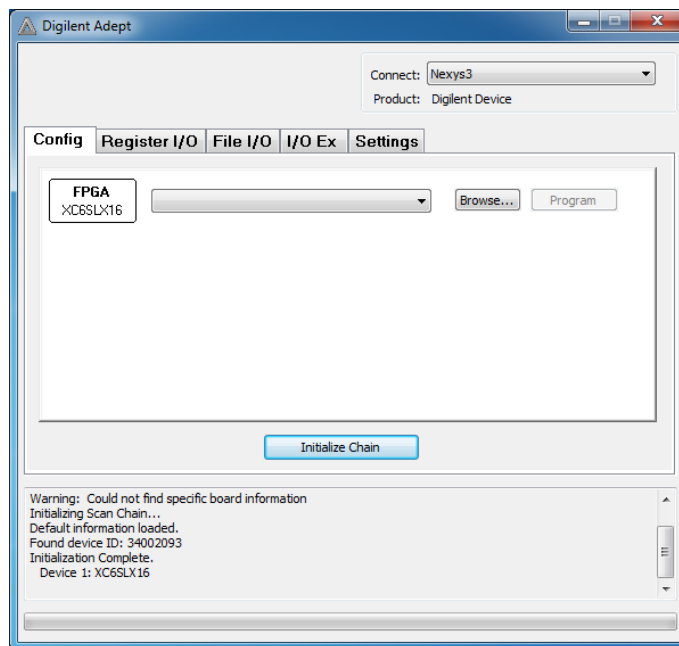
Po ukončení implementace se u každé části můžou zobrazit tři různé symboly, když byl nějaký krok úspěšný, tak se zobrazí symbol „fajfky“ (OK = v pořádku), . V případě nevhodného návrhu se zobrazí symbol vykřičník (warning = varování), toto varování neukončí implementaci, ale je vhodné si přečíst hlášení. Pokud se zobrazí symbol křížku (error = chyba), tak se ukončí aktuální krok implementace a je nutné chybu odstranit.



Obrázek 20: Symboly, a) „fajfka“, b) varování, c) chyba

V dalším kroku je potřeba vytvořit konfigurační soubor. Ten může mít různý formát v závislosti na konfiguraci. V tomto případě se vytvoří do složky s projektem soubor s příponou .bit. Soubor se vytvoří spuštěním „**Generate Programing File**“ který lze vidět na Obrázku 21.

Následuje konfigurace FPGA. K tomu je použit program Adept od firmy Digilent. Nejprve je nutné připojit vývojovou desku k počítači přes USB konektor (USB PROG) nacházející se v levém horním rohu desky. Nad tímto konektorem se nachází spínač pro zapnutí desky (červená LED dioda indikuje zapnutí). Dalším krokem je spuštění programu Adept. V pravém horním rohu se nachází záložka „**Connect**“, kde je vypsána deska Nexys 3. Program vypíše v dialogovém okně hlášení o inicializaci desky.



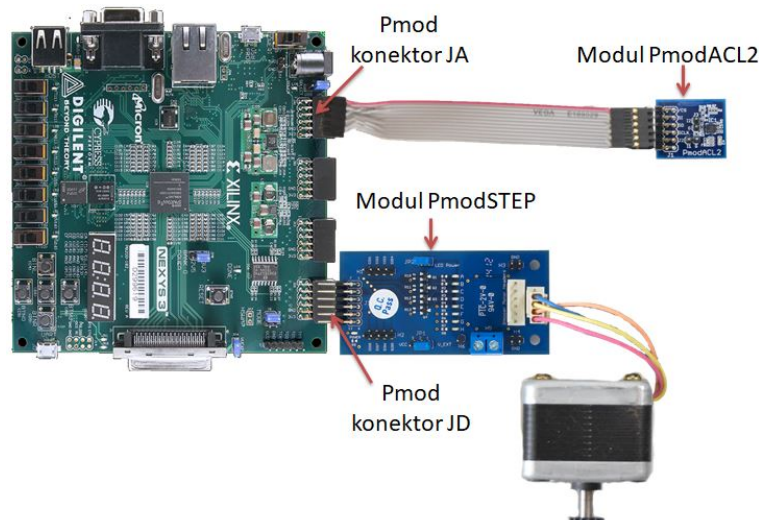
Obrázek 21: Program Digilent Adept

Poté se stiskne tlačítko „**Browse**“ a vyhledá se požadovaný soubor s příponou .bit. Poslední krok je stisknutí tlačítka „**Program**“, tím se nahraje do FPGA konfigurační soubor a v dialogovém okně se vypíše hlášení, zda je soubor úspěšně nahrán. V případě, že nahrání proběhlo v pořádku se vypíše „**Programming Successful**“.[1]

## B.6 Pracovní postup

### B.6.1 Postup k bodu č.1

1. Zapojte moduly do desky NEXYS 3 podle následujícího obrázku.



Obrázek 22: Celkový pohled na model

2. Spusťte program ISE Design Suite.



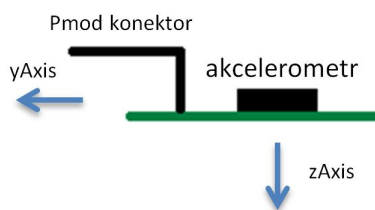
Obrázek 23: Ikona programu ISE Design Suite

3. Otevřete projekt Kyberneticky\_model.

- (a) File
- (b) Open Project
- (c) Otevřete složku projektu Kyberneticky\_model\_PmodACL2\_PmodSTEP
- (d) kyberneticky\_model\_PmodACL2\_PmodSTEP.xise

### B.6.2 Postup k bodu č.2

Vytvořte takovou podmínku, aby se při otočení modulu ( $+z \Rightarrow -z$ ) změnil směr otáčení hřídele krokového motoru, jak lze vidět na obrázku 24, osa z je kolmá na desku plošných spojů (DPS) modulu.

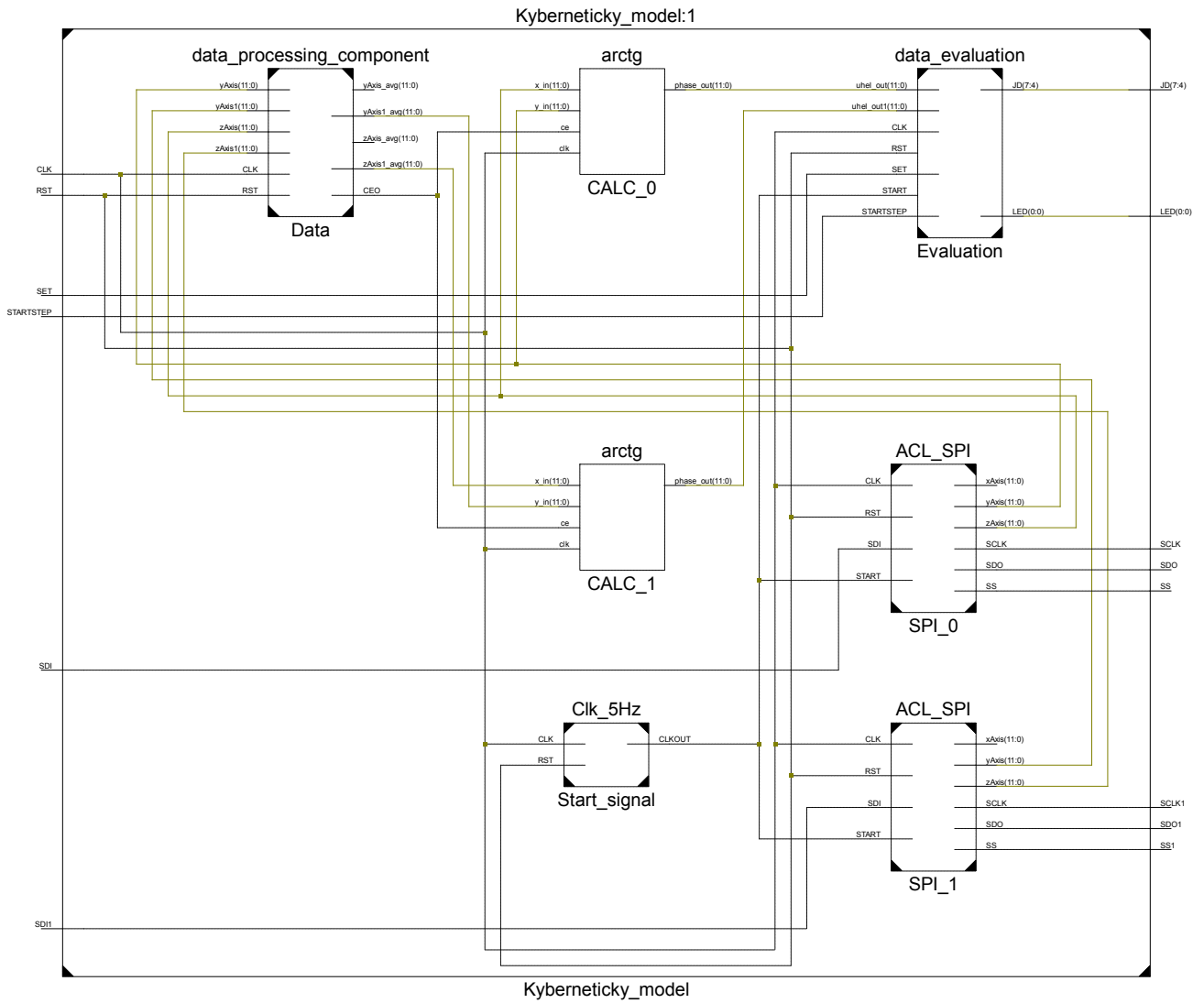


Obrázek 24: Pohled z boku na modul PmodACL2

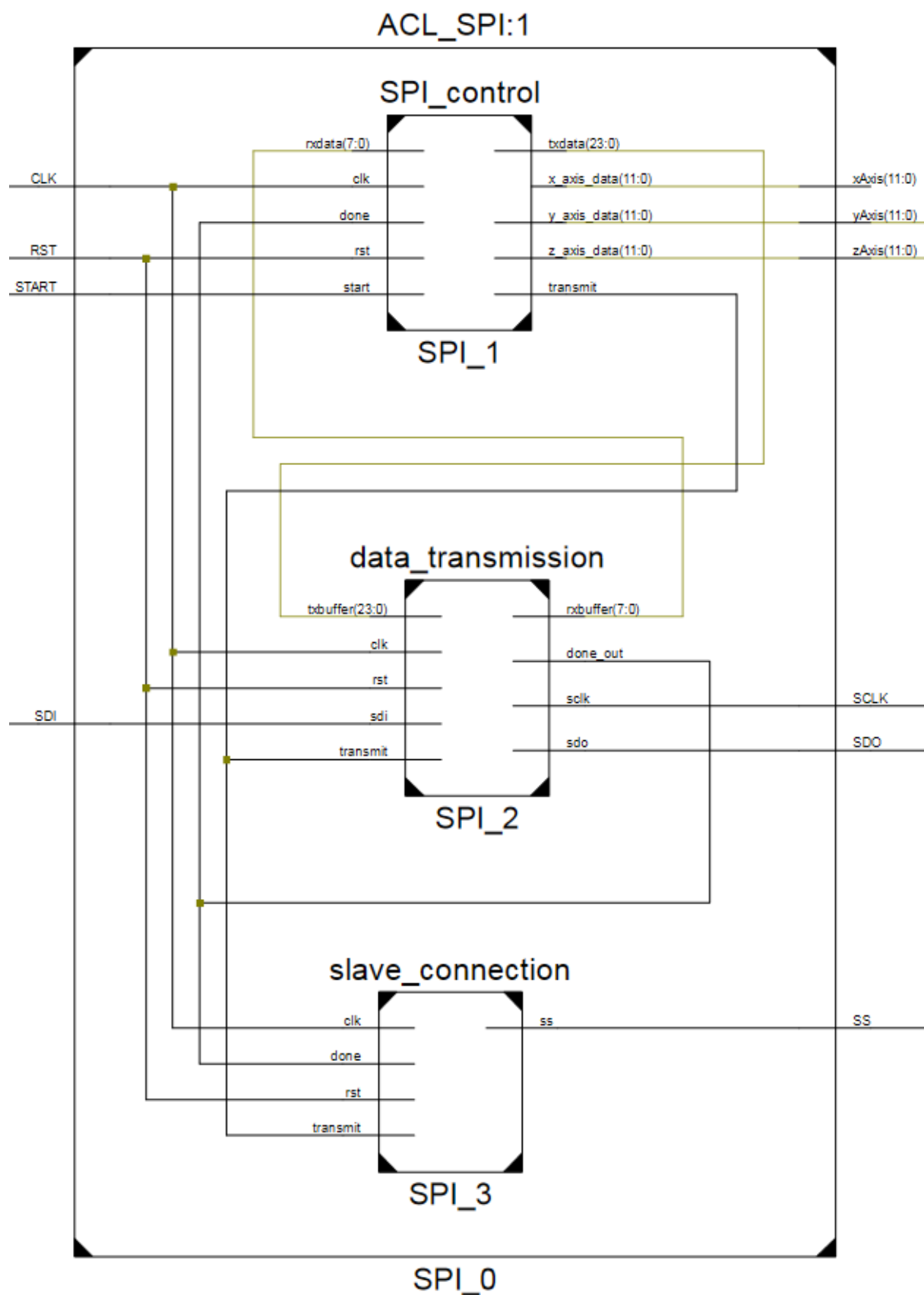
### Seznam použité literatury

- [1] HOLČÍK, Jiří, KOMENDA, Martin (eds.) a kol. Matematická biologie: e-learningová učebnice [online]. 1. vydání. Brno: Masarykova univerzita, 2015. ISBN 978-80-210-8095-9.
- [2] TREFIL, Adam. Výukový kybernetický model s krokovými motory – laboratorní úloha [online]. Ostrava, 2018 [cit. 2018-04-30]. Bakalářská práce. VŠB – Technická univerzita Ostrava Fakulta elektrotechniky a informatiky Katedra kybernetiky a biomedicínského inženýrství. Vedoucí práce Ing. Vladimír Kašík, Ph.D.

## C RTL schéma entity Kyberneticky model



## D RTL schéma komponenty ACL\_SPI



## Externí přílohy – Elektronické

### CD/DVD

- Bakalářská práce ve formátu .pdf
- Projektová složka bakalářské práce s názvem Kyberneticky\_model
- Projektová složka laboratorní úlohy PmodSTEP s názvem Kyberneticky\_model\_PmodSTEP
- Projektová složka laboratorní úlohy PmodSTEP s názvem Kyberneticky\_model\_PmodACL2\_PmodSTEP
- Video funkčního modelu